University of Bath

PHD

Mode jumping in MCMC

Behrens, Gundula

Award date:
2008

Awarding institution:
University of Bath

Link to publication

Alternative formats
If you require this document in an alternative format, please contact:
openaccess@bath.ac.uk

Copyright of this thesis rests with the author. Access is subject to the above licence, if given. If no licence is specified above, original content in this thesis is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC-ND 4.0) Licence (https://creativecommons.org/licenses/by-nc-nd/4.0/). Any third-party copyright material present remains the property of its respective owner(s) and is licensed under its existing terms.

Take down policy
If you consider content within Bath's Research Portal to be in breach of UK law, please contact: openaccess@bath.ac.uk with the details. Your claim will be investigated and, where appropriate, the item will be removed from public view as soon as possible.

Download date: 05. May. 2023

Mode Jumping in MCMC

submitted by

Gundula Ragna Behrens

for the degree of Doctor of Philosophy

of the

University of Bath

Department of Mathematical Sciences

March 2008

COPYRIGHT

Attention is drawn to the fact that copyright of this thesis rests with its author. A copy of this thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and they must not copy it or use material from it except as permitted by law or with the consent of the author.

This thesis may be made available for consultation within the University Library and may be photocopied or lent to other libraries for the purposes of consultation.

Signature of Author .....

Gundula Ragna Behrens

To my parents

Summary

Markov chain Monte Carlo (MCMC) methods often have difficulties in moving between isolated modes. To understand these difficulties, some MCMC theory and some mode jumping approaches will be reviewed, first in fixed dimension and later in variable dimension. The focus will lie on improving the efficiency of the powerful, but computationally expensive method "tempered transitions". A technique for optimising the method's parameters ("temperatures") will be proposed. It will be demonstrated that the default choice of geometric temperatures can be far from optimal. The tuning technique will then be tested on a hard applied sampling problem, namely on sampling from a fixed-dimensional mixture model. The results will show that the optimisation is robust and performs well and that tempered transitions achieves mode jumping ("label-switching") where standard MCMC fails. Since mixture models are often of variable dimension, it will be verified that tempered transitions and the tuning technique can also be applied in variable-dimensional problems. Tests on a variable-dimensional mixture model will confirm that tempered transitions also improves jumps between dimensions.

Acknowledgements

I am very grateful to my supervisor Dr Merrilee Hurn for introducing me to MCMC, for suggesting working on the compelling mode jumping problem and for supporting my endeavours with her wide knowledge, weekly feedback and constant encouragement. Her advice has been invaluable.

Also, I am deeply indepted to Evangelisches Studienwerk, EPSRC and the Department of Mathematical Sciences, University of Bath, for all their great help and support.

Finally, I thank my parents and brothers for being there for me and for believing in me. I am dedicating this thesis to my parents because they have helped me in making this dream (and many others) come true.

Contents

Chapter 1

Research Overview

1.1 Introduction

The work on "Mode Jumping in MCMC" is motivated by the indispensability of Markov chain Monte Carlo (MCMC) methods for the analysis of complex statistical models, which arise in a wide area of applications, for example in image analysis (e.g. in restoring blurred images), in spatial-temporal modelling (e.g. in simulating climate changes) or in cluster analysis (e.g. in identifying the genes in charge of the immune system). Due to the complexity, we can often only gain insight into the model by simulating its behaviour on the computer. A very common way of stochastic simulation – and often the only way – is to use MCMC methods.

Overall, MCMC is a very flexible and constantly growing class of methods. Each member of the class, however, has its flaws, which will show in certain cases. These shortcomings may lead to severe errors in the statistical inference. Therefore, it is essential to learn about what the drawbacks are and when they might arise. This knowledge can then be used to safeguard against such errors, for example by choosing an appropriate MCMC method and by monitoring its behaviour, so that MCMC performs well in many situations. The good experience encourages researchers to stretch the potential of MCMC either by fighting imperfection or by conquering new areas of application.

In this spirit, the following research on "Mode Jumping in MCMC" is dealing with one of the severe shortcomings, the lack of mode jumping in a multimodal environment, to which many MCMC methods are prone. The research overview starts with a crude description of the mode jumping problem, it then

guides through the course of investigation by a motivation-oriented summary of the main issues and results.

1.2 Lack of mode jumping in MCMC and its consequences

In statistics, being interested in a model means being interested in the associated probability distribution, which we will call $p(x)$. Often the only way to learn about the model, or more precisely about the distribution $p(x)$, is to generate and analyse samples from this distribution. In complex problems, these samples are produced by MCMC simulation.

MCMC offers an indirect way of sampling from a distribution $p(x)$: it generates states from a different random process, namely from a Markov chain, which converges to the desired distribution in equilibrium; due to this convergence property, the states of the Markov chain can be considered samples from the desired distribution, once the Markov chain has reached its equilibrium.

To illustrate how MCMC works, we will describe a very basic method for sampling from a distribution $p(x)$ defined on $\mathbb{R}^d$. This method constructs a Markov chain, which moves from state to state by the following iterative rule: if the current state of the Markov chain is $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, a proposal state $x' = (x'_1, \ldots, x'_d) \in \mathbb{R}^d$ is drawn from a multivariate normal distribution centred at the current state, i.e. $x' \sim N_d(x, \Sigma)$. A random decision will then determine the next state of the Markov chain: the next state will be either the proposal state x' $\left(\text{with probability } \alpha(x, x') = \min\left\{1, \frac{p(x')}{p(x)}\right\}\right)$ or the current state x. If the proposal state is chosen, we say that the proposal has been accepted, otherwise that it has been rejected. The form of the acceptance probability $\alpha(x, x') = \min\left\{1, \frac{p(x')}{p(x)}\right\}$ shows that a proposal state of higher probability than the current state (i.e. $p(x') > p(x)$) is always accepted, while a proposal state of a much smaller probability than the current state (i.e. $p(x') \ll p(x)$) is almost always rejected. In consequence, the Markov chain seeks high-probability areas (the modes of the distribution) and shuns low-probability areas.

If proposal states frequently land in low-probability area, the Markov chain keeps rejecting proposals and thus hardly moves. In that case, the number of

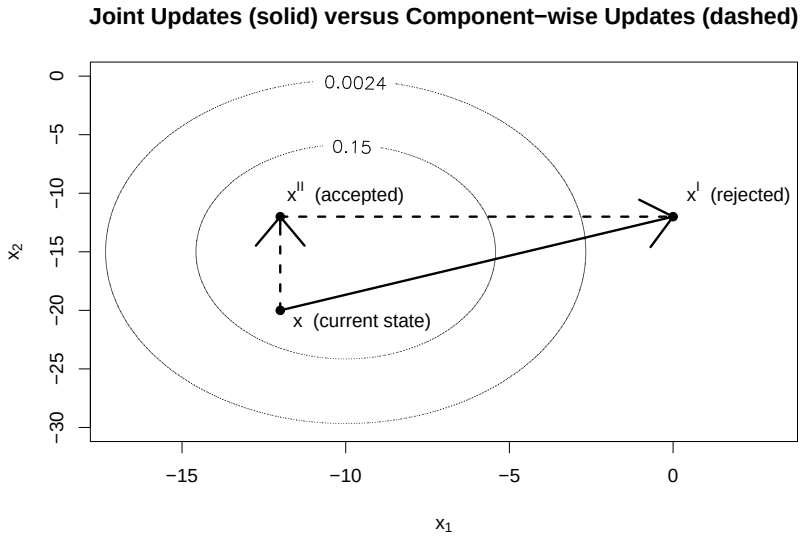

Figure 1-1: Suppose the chain currently visits the state $x = (x_1, x_2)$. A joint proposal $x' = (x'_1, x'_2)$ will lie in low-probability area if it contains at least one "badly" fitted component, say x'_1, in the sense that the hypothetical state $x'' = (x_1, x'_2)$ would be in high-probability area. Such a proposal x' is very likely to be rejected so that the chain remains in the current state x. Replacing the joint proposal by one sweep of component-wise proposals increases the probability that at least some movement takes place: first the move x to x'' may be considered and with high probability accepted; if x'' is accepted, then the move from x'' to x' may be proposed and with high probability rejected, so that the chain is very likely to end up in x'' after the sweep.

iterations needed for convergence of the method may lie far beyond feasibility. In particular, if the dimension d of the problem is high and all components are updated jointly, the probability that a proposal lies in low-probability area is high.

For the proposal's landing in low-probability area, it suffices that one of its components, say x'_1, is unlikely or badly fitted in the sense that the state without this component, i.e. the hypothetical state $x'' = (x_1, x'_2, \ldots, x'_d)$ lies in high-probability area. This is illustrated in Figure 1-1 for the two-dimensional case $d = 2$. Unfortunately, along with the rejection the information is also lost: in standard MCMC, there is no way of telling the algorithm to substitute the "badly" fitted component x'_1 in the proposal state $x' = (x'_1, \ldots, x'_d)$ by the well fitted component x_1 of the current state x and then to try again; such a modified state $x'' = (x_1, x'_2, \ldots, x'_d)$ would have a better chance of acceptance.

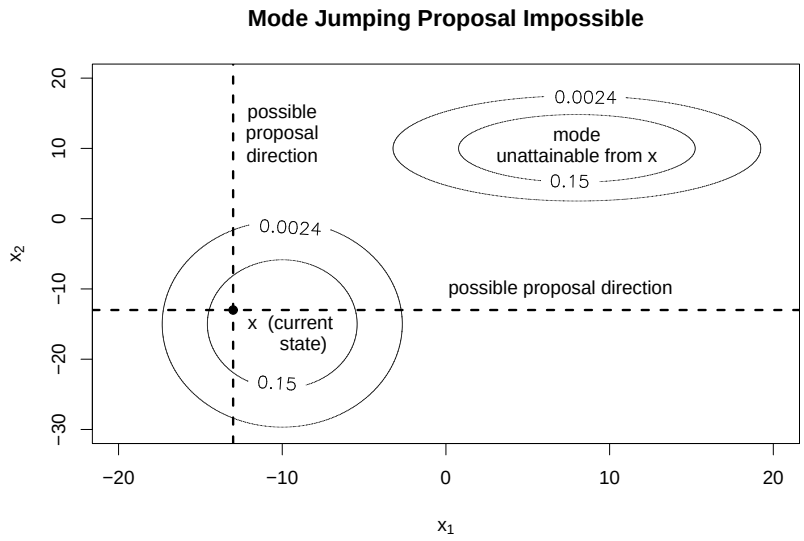

Figure 1-2: The drawback of component-wise updating is that modes outside the proposal directions, which are fixed by the current state x, cannot be attained.

What we can do, however, is to propose changes in only one component at a time. The proposal mechanism is then similar: if we are currently in $x = (x_1, x_2, \ldots, x_d)$, a new value for one of the components, say the first component, is drawn from the corresponding univariate normal distribution $x_1' \sim N(x_1, \sigma^2)$. The proposal state is then set equal to $x' = (x_1', x_2, \ldots, x_d)$ and accepted with probability $\alpha(x, x') = \min\left\{1, \frac{p(x')}{p(x)}\right\}$ as before. Markov chains based on such a component-wise update tend to have much higher acceptance rates and often mix well within one mode. However, component-wise updates limit the flexibility of the Markov chain since it can only move along the fixed directions. Within a single mode, this is usually not a serious restriction since every part of the single mode can be reached by a combination of moves along the fixed directions. Mixing between modes, however, will be impossible in the special case that the modes are separated by low-probability area and do not lie in a line with one of the fixed directions. Moving between such modes would require changing several components, which can neither be done at once (by definition of the component-wise update) nor in a sequence of component-wise moves since at least one of these sequential moves would lead into the low-probability area, which separates the modes, and thus be rejected (see Figure 1-2). Unfortunately, the alternative strategy of updating components jointly, which is able to jump between such modes at least in theory, does not perform any better; for changing more than one component at once increases the area

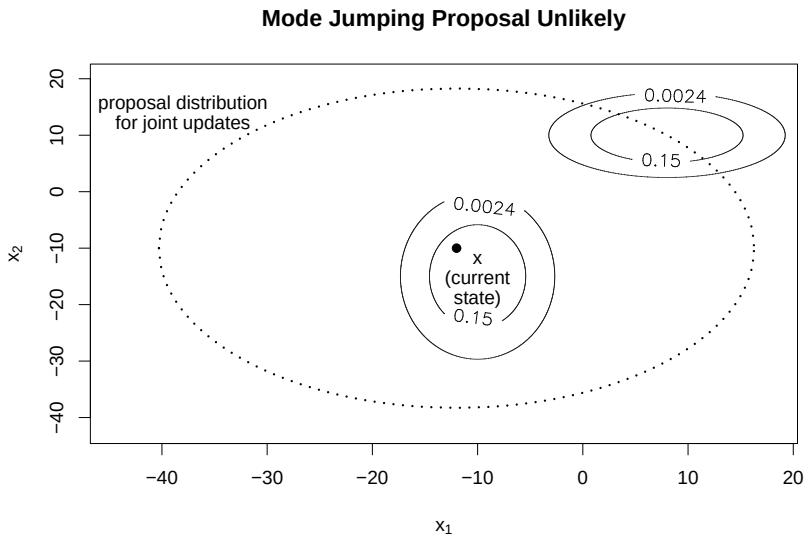

Figure 1-3: When updating components jointly, the area of possible proposal states may be so large that the probability of finding another mode is very small.

of possible proposal states substantially so that the probability of proposing a state from the other mode will be small. This is demonstrated in Figure 1-3 for the two-dimensional case. In this figure, the probability of proposing states from the other mode may still seem acceptable. However, this probability decreases with increasing dimension so that it will be unacceptably small in high-dimensional problems. In summary, standard MCMC mixes either poorly or not at all between modes.

If a Markov chain fails to visit some of the modes, it will convey incomplete information about the distribution of interest. As a result, any statistical inference based on this information will be incorrect. A complication is that it is hard to detect the lack of mode jumping because we often know too little about the number and location of modes to be able to verify that all modes have been visited. So far, poor mixing between modes is usually discovered by comparing the behaviour of several Markov chains which converge to the same equilibrium distribution. This is part of the convergence diagnostic for Markov chains. If all these chains are trapped in different modes of the distribution, it is obvious that the MCMC method is not mixing at all and that convergence has not taken place. Otherwise, if all the chains visit the same modes, good mixing and convergence of the MCMC method are assumed. Unfortunately, this assumption may be false; it might as well be that none of

the chains was able to discover the remaining modes of the distribution. Since the convergence diagnostic depends heavily on the MCMC method, it is only as reliable as this method and thus of limited help in the monitoring of MCMC. It is therefore inevitable to tackle the mixing problem directly by developing the mode jumping ability of MCMC further. The following research on "Mode Jumping in MCMC" is dedicated to this task. It aims to find and improve MCMC methods which can discover all the (possibly unknown) modes of a distribution and also jump frequently between them. Once this is achieved, a second step would be to find a reliable diagnostic which can tell whether all modes have been discovered. Developing such a diagnostic is not attempted here; it would be a worthy topic for future research.

1.3 Achieving and improving mode jumping in MCMC

1.3.1 Investigating mode jumping

The following investigation of "Mode Jumping in MCMC" consists of four parts: the first part explores the lack of mode jumping and possible remedies in problems of fixed dimension (Chapters 2 to 4). The second part improves the powerful, but expensive mode jumping method "tempered transitions" by developing a cost-reduction technique (Chapters 5 to 7). The third part extends the improved tempered transitions method to problems of variable dimension (Chapters 8 and 9). The final part closes the investigation by summarising and discussing the results and by suggesting areas of further research (Chapter 10).

The first part starts with an introduction to general MCMC and its alternatives (Chapter 2). The focus will lie on theory and construction of MCMC methods. This is necessary to fully understand the nature of the mode jumping problem. The ideas behind existing mode jumping approaches are then reviewed and their qualities assessed (Chapter 3). One of the promising mode jumping methods "tempered transitions" uses ideas from stochastic optimisation to slowly expand and then gradually contract the basin of attraction of each mode. This procedure allows the MCMC sampler to escape from the current mode and then to climb a new one. A similar approach is taken by another MCMC method "mode jumping via local optimisation". This method enables

mode climbing by deterministic rather than stochastic optimisation. Both methods, mode jumping via local optimisation and tempered transitions, have advantages and disadvantages so that it is not clear which method to prefer. So far, no comparative studies have been carried out. To fill this gap, tempered transitions and mode jumping via local optimisation are tested on a toy example (Chapter 4). The comparison shows that tempered transitions is not only the less expensive, but also the better mixing method. Despite being less expensive, tempered transitions still comes at a very high computational cost. So far, it is recommended to reduce the cost by tuning the method by trial and error. As this procedure may be tedious, a more efficient way of tuning is desirable. Finding such a way will fill the second part of this investigation.

1.3.2 Improving mode jumping in tempered transitions

The second part deals with the cost-efficiency of tempered transitions. Tempered transitions achieves jumps between modes of the distribution of interest via an auxiliary path generated under auxiliary distributions. Here a very common class of tempered transitions methods is investigated. This class samples from a distribution of the form $p(x) \propto \pi(x) \exp\left[-\beta_0 h(x)\right]$ where the parameter β_0 is usually set equal to one by excursions over its tempered versions $p_{\beta_i}(x) \propto \pi(x) \exp\left[-\beta_i h(x)\right]$, $i = 0, \ldots, n$, each of which is characterised by the so-called inverse temperature $\beta_i \in [0, 1]$. The function $h(x)$ is called the energy function.

Under idealising assumptions, the efficiency of the method depends solely on the temperature scheme $\{\beta_i\}_{i=0}^n$. For a given number of temperatures n, the true tuning problem is to find a temperature sequence between the smallest value $\beta_{\min}$ and the target temperature β_0 that maximises the expected acceptance probability of the algorithm. In most cases, the true problem is intractable. But we can tackle it implicitly by the related, albeit not equivalent problem of minimising the "sum of squares"

$$S\left(\{\beta_i\}_{i=0}^n\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i)\right]$$

where

$$g(\beta) := \mathbb{E}_{p_\beta}\left[h(X)\right].$$

We will explain how these problems are connected and how the related problem can be solved if the curve $g(\beta)$ is known (Chapter 5). To get a feeling for

the closeness of the two optimisation problems, we will compare the optimal solutions in the special case of sampling from the simplified Witch's Hat distribution, in which both problems are tractable (Chapter 6). Since the solution depends on the shape of the curve $g(\beta)$, we will cover the range of shapes that the curve can take by looking at various Witch's Hat examples. In all examples, there is little difference between the two optimal solutions; it seems that the related optimisation problem is not a bad approximation to the true optimisation problem. Originally, it was recommended to use a geometric temperature scheme. We will demonstrate that this scheme can be far from optimal. We will also derive criteria for assessing the appropriateness of a certain scheme. Furthermore, we will show that carrying out the optimisation under idealising assumptions also improves the efficiency of tempered transitions if these assumptions are not met. Encouraged by these results, the optimisation technique is extended so that it can also be applied to complex problems, where $g(\beta)$ is not analytically available. In this case, the key idea is to estimate $g(\beta)$ for some β values by importance sampling and then to interpolate $g(\beta)$ for all the remaining β values; based on this interpolation, the optimisation technique can be carried out as before (Chapter 7). Finally, the extended technique and its robustness are tested on a complex sampling problem, namely on sampling from a fixed-dimensional mixture of distributions, which model the well-known "galaxy data". The results are very satisfying. They show not only that the extended optimisation technique is robust and performing well, but also that the method of tempered transitions achieves mode jumping between modes (caused by the so-called "label switching"), while standard MCMC methods fail.

1.3.3 Tempered transitions in variable dimension

Since tempered transitions is a powerful mode jumping method, the third part of this investigation discusses its application to complex problems of variable dimension. First, the standard trans-dimensional MCMC method "reversible jump MCMC" is introduced with discussion of theory, mixing problems and further developments (Chapter 8). Then the validity of combining RJMCMC with tempered transitions is verified, before its application is tested on the variable-dimensional mixture model of the "galaxy" data (Chapter 9). In the "galaxy" example, both standard RJMCMC and variable-dimensional tempered transitions perform satisfactorily. However, tempered transitions is again more efficient.

1.3.4 Conclusions

The fourth part (Chapter 10) will discuss this work on "Mode Jumping in MCMC" and point out possible directions of future research. As we have already summarised the research above, we will now only mention the ideas for further research. We have learnt that tempering methods are good at mode jumping and that the tuning technique works well for tempered transitions. Hence, it seems rewarding to investigate whether similar tuning techniques can be applied to optimise other tempering methods. We have also learnt that basing convergence diagnostics on the MCMC method under investigation leads to unreliable diagnostics because this method may not have visited a certain region at all. Due to the better mixing at higher temperatures, it may be possible to develop a reliable convergence diagnostic based on the information gained at a hot temperature. For example, if there is a variable whose empirical mean only converges to the (unknown) theoretical mean when the sampler is visiting all the modes, we could estimate the true mean by importance sampling (based on a sample from the hot distribution) and compare this estimate with the one obtained by normal MCMC estimation (based on a sample from the target distribution). If both estimates approximately agree, we can infer that convergence has taken place.

Chapter 2

MCMC Theory

2.1 Introduction

In statistical inference, the problem of evaluating expected values of the form $\mathbb{E}_\pi[h(X)] = \int h(x)\,\pi(x)\,\mu(\mathrm{d}x)$ is omnipresent (Section 2.2). It can often be tackled by Markov chain Monte Carlo (MCMC) estimation. MCMC is a sophisticated method which should only be applied if no simpler method such as Monte Carlo estimation (Section 2.3) is feasible. It is helpful to understand the complexity of MCMC before starting working with the method. The roots of the complexity lie in the theoretical foundation of MCMC. In MCMC, we set up a Markov chain which converges to the target distribution π in equilibrium (Section 2.4). If we run this Markov chain long enough, we can use its states $X_1, \ldots, X_N$ for estimating $\mathbb{E}_\pi[h(X)]$ by the empirical average $\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(X_i)$ (Section 2.5). The accuracy of the estimator depends very much on the mixing properties of the chain. The art of MCMC lies therefore in its construction. The generality of the method offers a great freedom of design. Among the various MCMC methods, some standard methods (Metropolis-Hastings algorithm and Gibbs sampler) have been proven helpful either on their own or as part of a more sophisticated MCMC method (Section 2.6). The design is not the only worry when implementing MCMC. A very important issue is how to diagnose the convergence of the chain (Section 2.7). Another issue is how many iterations the chain needs to find the modal area of the distribution. Discarding these iterations improves the accuracy of the estimation (Section 2.8). In the past, there was also much dispute on whether the estimation should be based on the output of a single chain or on the pooled output of several chains (Section 2.9).

2.2 Expectations of random variables

Many problems which we face in statistical inference can be reduced to the problem of obtaining the expected value of some function of random variables. A common example is determining the mean or the variance. To define the expectation of a random variable X or of its function $h(X)$, we have to introduce some notation. Recall that, in statistics, the random variable X is defined on the σ-finite measure space $(\Omega, \mathcal{A}, \mu)$. The measure space consists of the sample space (or state space) Ω, the σ-finite measure μ and the σ-algebra $\mathcal{A}$, which contains all measurable sets $A \subset \Omega$. For instance, if the sample space Ω is discrete (e.g. $\Omega = \mathbb{Z}^d$), then the measure μ is the counting measure; if Ω is continuous (e.g. $\Omega = \mathbb{R}^d$), then μ is the Lebesgue measure. On this space, the random variable is defined by a distribution or equivalently by a density. It is common to simplify notation by using the same expression for both distribution and density. In the following, we will denote the distribution and density of X by π. This enables us to express the expected value $\mathbb{E}_\pi[h(X)]$ by the integral

$$\mathbb{E}_\pi[h(X)] = \int_\Omega h(x)\,\pi(x)\,\mu(\mathrm{d}x).$$

If $h(x)$ is an indicator function, the expected value represents a probability: let

$$\mathbb{1}_A(x) = \begin{cases} 1 & \text{if } x \in A, \\ 0 & \text{otherwise,} \end{cases}$$

be the indicator function that x lies in the set A, then the expectation $\mathbb{E}_\pi[\mathbb{1}_A(X)]$ describes the probability $\mathbb{P}\{X \in A\}$ that X lies in A, for

$$\begin{aligned} \mathbb{E}_\pi[\mathbb{1}_A(X)] &= \int_\Omega \mathbb{1}_A(x)\,\pi(x)\,\mu(\mathrm{d}x) \\ &= \int_A \pi(x)\,\mu(\mathrm{d}x) \\ &= \mathbb{P}\{X \in A\}. \end{aligned}$$

In simple cases, we can evaluate $\mathbb{E}_\pi[h(X)]$ by analytical or numerical integration. If this is not feasible, another possibility is to estimate the expectation by the empirical mean over a sample generated by Monte Carlo or Markov chain Monte Carlo simulation. Let us start with the simpler method of the two, Monte Carlo estimation.

2.3 Monte Carlo estimation

The most common way of Monte Carlo estimation is to draw exact and independent samples $X_1, X_2, \ldots, X_N$ from the distribution $\pi(x)$ and to plug them into the estimator

$$\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i),$$

which converges to $\mathbb{E}_\pi [h(X)]$ as N goes to infinity. Due to the independence between samples, the convergence of this estimator can be verified by the law of large numbers and the central limit theorem (for example given in Grimmett and Stirzaker 2004, Section 5.10): by the law of large numbers, the estimator converges in distribution to the expectation

$$\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i) \quad \xrightarrow{\mathcal{D}} \quad \mathbb{E}_\pi [h(X)] \qquad \text{as} \quad N \to \infty;$$

moreover, if the variance $\mathrm{var}_\pi [h(X)]$ is finite, then, by the central limit theorem, the Monte Carlo error is asymptotically normally distributed,

$$\sqrt{N} \left\{ \bar{h}_N - \mathbb{E}_\pi [h(X)] \right\} \quad \xrightarrow{\mathcal{D}} \quad N\big(0, \mathrm{var}_\pi [h(X)]\big) \qquad \text{as} \quad N \to \infty. \tag{2.1}$$

For generating samples from $\pi(x)$, a variety of techniques is employed including pseudo-random-number generators, transformation of variables (i.e. inversion), rejection sampling and other distribution-specific approaches. Most methods yield independent samples; some methods, however, deliberately induce negative autocorrelations (and thus dependencies) between samples ("antithetic variables") and thus achieve a smaller variance in estimation than independent sampling methods do. Many of these sampling methods, including algorithms for standard distributions, are described in Ripley (1987). Some additional advice regarding the simulation of standard random variables can be found in Gelman, Carlin, Stern and Rubin (2004, Appendix A).

Another way of estimation is importance sampling. We use it if we cannot easily sample from the distribution π, but are able to produce samples from an over-dispersed distribution ψ, which covers the modes and tails of π. Although importance sampling originates in Monte Carlo estimation, it also works when the estimation is based on MCMC samples. Importance sampling uses a "trick" to estimate the desired expectation. It expresses the original expectation $\mathbb{E}_\pi [h(X)]$, which is given with respect to π, by the ratio of two different

expectations, each with respect to the "importance sampling distribution" ψ:

$$\mathbb{E}_\pi\left[h(X)\right] = \frac{\int_\Omega h(x)\,\frac{\pi(x)}{\psi(x)}\,\psi(x)\,\mu(\mathrm{d}x)}{\int_\Omega \frac{\pi(x)}{\psi(x)}\,\psi(x)\,\mu(\mathrm{d}x)} = \frac{\frac{\int_\Omega h(x)\,\frac{\pi(x)}{\psi(x)}\,\psi(x)\,\mu(\mathrm{d}x)}{\int_\Omega \psi(x)\,\mu(\mathrm{d}x)}}{\frac{\int_\Omega \frac{\pi(x)}{\psi(x)}\,\psi(x)\,\mu(\mathrm{d}x)}{\int_\Omega \psi(x)\,\mu(\mathrm{d}x)}} = \frac{\mathbb{E}_\psi\left[\frac{\pi(X)}{\psi(X)}\,h(X)\right]}{\mathbb{E}_\psi\left[\frac{\pi(X)}{\psi(X)}\right]}$$

where π and ψ may be unnormalised. This expression shows that the expectation can be estimated by generating $y_1, y_2, \ldots, y_N$ from $\psi(y)$ (by Monte Carlo simulation or MCMC) and plugging these values into

$$\widehat{\mathbb{E}_\pi\left[h(X)\right]} = \frac{\widehat{\mathbb{E}_\psi\left[\frac{\pi(X)}{\psi(X)}\,h(X)\right]}}{\widehat{\mathbb{E}_\psi\left[\frac{\pi(X)}{\psi(X)}\right]}} = \frac{\frac{\sum_{i=1}^N \frac{\pi(y_i)}{\psi(y_i)}\,h(y_i)}{N}}{\frac{\sum_{i=1}^N \frac{\pi(y_i)}{\psi(y_i)}}{N}} = \frac{\sum_{i=1}^N w_i\,h(y_i)}{\sum_{i=1}^N w_i}, \qquad w_i = \frac{\pi(y_i)}{\psi(y_i)},$$

(see for example Hastings 1970). The ratios $w_i = \frac{\pi(y_i)}{\psi(y_i)}$, $i = 1, \ldots, N$, are called importance weights. The better the over-dispersed distribution ψ approximates the desired distribution π, the robuster is this estimation.

2.4 Markov chains for MCMC estimation

2.4.1 Markov chains

We will briefly discuss the key ingredients of MCMC simulation. For a thorough probabilistic review, see for example the introduction to finite space Markov chains with a section on MCMC in Grimmett and Stirzaker (2004, Chapter 6) or the discussion of general state space MCMC in Tierney (1994), in Tierney (1996) and in Robert and Casella (1999, Chapters 4 and 6). The latter sources often refer to Markov chain theory presented in Nummelin (1984).

A Markov chain $\{X_t\}_{t\in\mathbb{N}_0}$ is a discrete-time random process, which can be best described by its transition kernel. A (Markov) transition kernel is a map $P:\ \Omega \times \mathcal{A} \to [0,1]$, which is defined in such a way that, for every $x \in \Omega$, the function $P(x,\cdot)$ is a probability measure, and for every $A \in \mathcal{A}$, the function $P(\cdot, A)$ is measurable. In particular, the Markov property means that

$$\begin{aligned} P(X_t, A) &= \mathbb{P}\left\{X_{t+1} \in A | X_0, \ldots, X_{t-1}, X_t\right\} \\ &= \mathbb{P}\left\{X_{t+1} \in A | X_t\right\}. \end{aligned}$$

In other words, the next state X_{t+1} of the Markov chain only depends on the current state X_t, but not on any of the past states $X_0, \ldots, X_{t-1}$. The next state X_{t+1} is a random variable, which follows a probability distribution conditional on the current state X_t. If we start the Markov chain in the state X_0, the

conditional distribution of the current state X_t given the initial state X_0 is given by

$$\mathbb{P}\{X_t \in A | X_0\} = P^t(X_0, A),$$

where P^t denotes the tth iterate of the kernel P.

Markov chains used in MCMC are π-invariant, π-irreducible and aperiodic. These notions shall be explained in the following sections.

2.4.2 Invariance

A Markov chain is invariant if

$$\pi(A) = \int \pi(\mathrm{d}x) P(x, A) \qquad \text{for all } A \in \mathcal{A} \quad \text{(global balance)}.$$

The most convenient way of achieving invariance is to construct a reversible Markov chain because reversibility can be easily verified. A Markov chain is reversible if the "detailed balance" condition holds. In the literature, both of the following detailed balance definitions can be found: a Markov chain satisfies the detailed balance condition if

$$\pi(x)P(x, \mathrm{d}x') = \pi(x')P(x', \mathrm{d}x) \qquad \text{for all } x, x' \in \Omega \quad \text{(detailed balance)}$$

or if

$$\int_A \int_B \pi(\mathrm{d}x) P(x, \mathrm{d}x') = \int_B \int_A \pi(\mathrm{d}x') P(x', \mathrm{d}x) \qquad \text{for all } A, B \in \mathcal{A} \quad \text{(detailed balance)}.$$

2.4.3 Irreducibility

A Markov chain is called π-irreducible if it is able to visit all sets that have positive probability under π. More formally, π-irreducibility says that, for every initial state $x \in \Omega$ and for every set $A \in \mathcal{A}$ with $\pi(A) > 0$, there exists a time $t = t(x, A) > 0$ such that the probability $P^t(x, A) > 0$ of visiting the set A at time t given the starting point x is positive. The value of this time $t = t(x, A)$ may depend on the starting point x and the set A. Otherwise, the Markov chain is reducible, which means that eventually it will be trapped forever in one part of the sample space.

In the statistics literature, an irreducible Markov chain is sometimes called "nearly reducible" if the probability of escaping from a (mode) trap is very small.

2.4.4 Aperiodicity

A π-irreducible Markov chain is aperiodic if it does not move through the sample space in a cyclic manner. A d-cycle is a sequence of d non-empty disjoint sets $\{A_0, A_1, \ldots, A_{d-1}\}$, through which the chain always passes in the same order and out of which it cannot escape because the probability of moving to the next set is equal to one. If the chain currently visits the set A_i and if this set is not the last in the cycle, then the Markov chain will jump to the next set A_{i+1} with probability one (i.e. $P(x, A_{i+1}) = 1$ for all $x \in A_i$, $i = 0, \ldots, d-2$); otherwise, if it visits the last set A_{d-1}, it will return to the very first set A_0 also with probability one (i.e. $P(x, A_0) = 1$ for all $x \in A_{d-1}$). If there exists such a d-cycle of length greater than one ($d > 1$), then the chain is called periodic with period d, otherwise aperiodic.

For instance, the simple symmetric random walk on $\mathbb{Z}$ is periodic with period $d = 2$. This random walk is structured as follows: if the Markov chain is in $x \in \mathbb{Z}$, it will either move to $x + 1$ or to $x - 1$ with equal probability $P(x, x+1) = P(x, x-1) = \frac{1}{2}$ in the next iteration. If it currently visits an odd number, it is not able to visit the same or another odd number in the next iteration; similarly, it can not move between even numbers within one iteration. Let us denote the set of even numbers by $A_0 := \{2\,j : \ j \in \mathbb{Z}\}$ and the set of odd numbers by $A_1 := \{2\,j + 1 : \ j \in \mathbb{Z}\}$, then the chain constantly moves between these two disjoint sets and thus has period $d = 2$. If we changed the structure of the random walk such that there is always the possibility of remaining in the current state, e.g. $P(x, x) = P(x, x+1) = P(x, x-1) = \frac{1}{3}$, we would obtain an aperiodic Markov chain.

2.4.5 Ergodicity

If a Markov chain is π-invariant, π-irreducible and aperiodic, it converges to the distribution π. In MCMC, this distribution is often called target distribution because it is the distribution from which we want to sample.

The convergence of a Markov chain to another distribution is often expressed in terms of the total variation distance $\|\cdot\|_{TV}$ defined by

$$\left\|P^t(x, \cdot) - \pi(\cdot)\right\|_{TV} := \sup_A \left|P^t(x, A) - \pi(A)\right|.$$

Note that there is also an alternative definition of the total variation distance, which differs from this one by a factor of 2.

For a π-invariant, π-irreducible and aperiodic Markov chain, we have that

$$\lim_{t\to\infty} \left\| P^t(x,\cdot) - \pi(\cdot) \right\|_{TV} \to 0, \qquad \text{for } \pi\text{-almost all } x \in \Omega.$$

From a theoretical point of view, it is a nuisance that convergence only takes place for π-almost all starting points. In practice, this does not matter because all MCMC samplers that can be run on a computer satisfy an additional condition "Harris recurrence" so that convergence from all starting points is ensured (Chan and Geyer 1994). Roughly speaking, Harris recurrence means that the Markov chain visits every relevant region of the sample space infinitely often. The formal definition is that, for each $B \subset \Omega$ with $\pi(B) > 0$,

$$\mathbb{P}\left\{X_n \in B \text{ infinitely often } | X_0 = x\right\} = 1 \qquad \text{for all } x \in \Omega.$$

If the Markov chain also satisfies Harris recurrence, it is ergodic, and the following statements are equivalent:

1. The Markov chain is ergodic.
2. There exists a probability distribution π such that
$$\lim_{t\to\infty} \left\| P^t(x,\cdot) - \pi(\cdot) \right\|_{TV} \to 0, \qquad \text{for all } x \in \Omega.$$
3. The Markov chain is π-invariant, π-irreducible, aperiodic and Harris recurrent.

In practice, checking invariance, irreducibility and aperiodicity suffices to verify ergodicity.

2.4.6 Mixing

This research deals with slowly mixing Markov chains in MCMC. This is an important research area because slow mixing causes slow convergence.

In general, mixing describes the lag t dependence structure of the Markov chain, i.e. the dependence between the states X_i and X_{i+t}. One way of measuring this dependence is the autocorrelation function

$$\rho(t) = \text{corr}(X_i, X_{i+t}), \qquad t = 0, 1, \ldots$$

A chain is called slowly/rapidly mixing if this dependence is slowly/rapidly decaying in t (Geyer 1992).

In the statistics literature, "mixing" and "convergence" are used almost interchangeably. This may be due to the closeness of both concepts: certain types of mixing imply certain types of convergence and vice versa (Tierney 1996, Robert 1994). For instance, exponential φ-mixing,

$$\varphi(t) := \sup_{A,B} |P_\pi(\, X_t \in A | X_0 \in B\,) - \pi(\, A\,)| = O(r^t),$$

is equivalent to uniform ergodicity,

$$\left\| P^t(x, \cdot) - \pi(\cdot) \right\|_{TV} \leq M r^t \quad \text{for all } x \text{ and all } t,$$

where M and r are constants with $M < \infty$ and $0 < r < 1$.

In addition, "mixing" is often used informally to describe the flexibility of a Markov chain. It is for example common to distinguish between the "mixing between modes" and the "mixing within modes" when depicting the qualities of an MCMC method. This informal notion is helpful to locate the source of slow mixing. For instance, a Markov chain may be very fast in exploring a single mode ("fast mixing within modes"), but hardly able to move between modes ("poor mixing between modes"). In this situation, improving the method's mode jumping ability will speed up the mixing (and thus the convergence) of the chain.

2.5 MCMC estimation

2.5.1 Justification

When using an MCMC method, we often have to check on a case-by-case basis that this MCMC method generates an ergodic Markov chain which converges to the distribution of interest. If this is verified, then the MCMC estimation can be justified by the following ergodic theorem: suppose the Markov chain $\{X_i\}_{i \in \mathbb{N}}$ is ergodic and has equilibrium distribution π; suppose further that the expectation with respect to the absolute value of the function $h(\cdot)$ is finite, $\mathbb{E}_\pi \{|h(X)|\} < \infty$; then the empirical average $\bar{h}_N = \frac{1}{N} \sum_{i=1}^{N} h(X_i)$, which is based on the states of the Markov chain, converges almost surely to the desired expectation $\mathbb{E}_\pi [h(X)]$ as N goes to infinity.

Furthermore, we know from time series and spectral analysis (Priestley 1981, Section 5.3.2), that, for large N, the variance of the empirical mean $\bar{h}_N$ is approximately

$$\operatorname{var}\left(\bar{h}_N\right) \sim \frac{1}{N} \operatorname{var}_\pi \left[h(X)\right] \tau(h)$$

where $\tau(h)$ denotes the integrated autocorrelation time

$$\tau(h) = \sum_{t=-\infty}^{\infty} \rho_t(h) = 1 + 2 \sum_{t=1}^{\infty} \rho_t(h)$$

and $\rho_t(h)$ denotes the autocorrelation of the chain $\{h(X_i)\}$ at lag t

$$\rho_t(h) = \rho_{-t}(h) = \frac{\gamma_t(h)}{\gamma_0(h)} \quad \text{where} \quad \gamma_t(h) = \gamma_{-t}(h) = \operatorname{cov}\left[h(X_i), h(X_{i+t})\right].$$

Hence, the estimator's variance is proportional to the integrated autocorrelation time $\tau(h)$.

A stronger convergence result (central limit theorem) holds for example if the Markov chain is ergodic and reversible (Geyer 1992, Theorem 2.1):

$$\sqrt{N}\left\{\bar{h}_N - \mathbb{E}_\pi\left[h(X)\right]\right\} \quad \xrightarrow{\mathcal{D}} \quad N\left(0, \operatorname{var}_\pi\left[h(X)\right] \tau(h)\right), \qquad \text{as} \quad N \to \infty. \tag{2.2}$$

A comparison between the central limit theorems (2.1) and (2.2) shows that the variance of the MCMC error differs from the variance of the Monte Carlo error by a factor of $\tau(h)$. Since $\tau(h)$ is usually greater than one, Monte Carlo estimation is more accurate and should be preferred whenever possible.

2.5.2 Measures of performance

The convergence result

$$\operatorname{var}\left(\bar{h}_N\right) \sim \frac{1}{N} \operatorname{var}_\pi \left[h(X)\right] \tau(h)$$

can be used to assess the accuracy of a method if the number of iterations N is fixed. For fixed N, the factor $\frac{1}{N} \operatorname{var}_\pi \left[h(X)\right]$ is a constant, albeit an unknown one because it cannot be determined analytically. This constant is the same for all methods so that, for each method, the quality of estimation is fully characterised by the integrated autocorrelation time $\tau(h)$. Hence, $\tau(h)$ can be used as a comparative measure for the accuracy of estimation under a particular method.

Suppose we compare two methods with each other. Then the one with the smaller integrated autocorrelation time is the more accurate one. If a method is

assessed on its own, we can judge how it performs in comparison to independent sampling. Independent sampling yields an integrated autocorrelation time equal to one since independence implies that $\rho_0(h) = 1$ and $\rho_t(h) = 0$ for all $t \neq 0$ so that $\tau(h) = \sum_{t=-\infty}^{\infty} \rho_t(h) = 1$. Hence, $\tau(h) = 1$ is the benchmark of the comparative measure. Usually, MCMC is less accurate ($\tau(h) > 1$) than independent sampling because samples tend to be positively autocorrelated. A higher accuracy ($\tau(h) < 1$) can only be achieved through strong negative autocorrelations between the MCMC samples. In general, methods producing such negative autocorrelations are rare; however, in some special cases, they exist [see Green and Han's (1992) method for example].

When comparing two methods, we have not only to consider the accuracy of estimation, but also the computational cost. Suppose one method is twice as accurate than the other, but takes ten times longer to generate the same sample size, then it is five times less cost-efficient so that the other method is usually preferred. Hence, a measure for cost-efficiency is computing time (for fixed N) times integrated autocorrelation time.

Usually, the integrated autocorrelation time $\tau(h)$ cannot be determined directly so that it has to be estimated. One has to be careful with the choice of estimator. The "natural" estimator $\hat{\tau}_{\text{nat}}(h) = 1 + 2\sum_{t=1}^{N-1} \hat{\rho}_t(h)$ is a bad choice because it is inconsistent, but alternative estimators exist (see for example Priestley (1981, Section 6.2.3)).

If the samples are produced by a reversible MCMC method, then a very good estimator for $\tau(h)$ is Geyer's (1992) initial positive sequence estimator. This estimator checks whether the sample autocorrelations behave as theoretically expected. In theory, the reversibility of the chain implies that the sum of two adjacent autocorrelations (starting at the even lag) is positive, $\rho_{2k} + \rho_{2k+1} > 0$ for all $k \in \mathbb{N}_0$. In practice, we can only check the behaviour of the estimated autocorrelations (sample autocorrelations). Since the sample autocorrelations $\hat{\rho}_t(h)$ are also prone to error which increases as the lag t increases, it will happen that $\hat{\rho}_{2k} + \hat{\rho}_{2k+1} \leq 0$ at some point. When it happens for the first time, the noise level of the sample autocorrelations is definitely reached. None of the sample autocorrelations beyond this point should be used for estimating the integrated autocorrelation time $\tau(h)$ because any estimator based on noise is unreliable. In consequence, the initial positive sequence estimator only includes

sample autocorrelations up to the first integer $(m+1)$ at which this deviation from theory occurs ($\hat{\rho}_{2(m+1)} + \hat{\rho}_{2(m+1)+1} \leq 0$).

More formally, the initial positive sequence estimator is given by

$$\hat{\tau}(h) = 1 + 2\sum_{t=1}^{M} \hat{\rho}_t(h) \tag{2.3}$$

where the "window-width" $M = 2m + 1$ is defined by

$$m = \max\left\{t \in \mathbb{N}_0 : \quad \hat{\rho}_{2k}(h) + \hat{\rho}_{2k+1}(h) > 0 \quad \text{for all} \quad k = 0, 1, \ldots, t\right\}.$$

The autocorrelations can be estimated by $\hat{\rho}_t(h) = \frac{\hat{\gamma}_t(h)}{\hat{\gamma}_0(h)}$ where

$$\hat{\gamma}_t(h) = \frac{1}{N}\sum_{i=1}^{N-t} \left[h(X_i) - \bar{h}_N\right]\left[h(X_{i+t}) - \bar{h}_N\right]$$

as recommended in Priestley (1981, Sections 5.3.3 and 5.3.4).

It is sometimes suggested to reduce correlation between samples and thus to improve the integrated autocorrelation time by "subsampling" (or "thinning") the chain, i.e. by taking only every kth sample. Weighing the gain against the computational cost, it is usually best to take the entire sample (i.e. $k = 1$); only in a few cases, k less than five is appropriate (Geyer 1991, Geyer 1992). In practice, the sample is sometimes thinned to reduce the cost of storage or the computational cost if calculating the quantity of interest is very expensive.

2.6 Standard MCMC methods

2.6.1 Metropolis-Hastings algorithm

In general, any method which produces a π-invariant, π-irreducible and aperiodic Markov chain can be used to sample from the distribution π and to estimate any expectation $\mathbb{E}_\pi[h(X)]$. However, there are some standard methods, which shall be introduced here.

The most common MCMC method is the Metropolis-Hastings algorithm (Hastings 1970). Given an initial state X_0, it generates a reversible Markov chain with invariant distribution π in an iterative manner. At each iteration $t = 1, 2, \ldots, N-1$, it proceeds as follows:

Algorithm 2.1:

Step 1 Set $x = X_{t-1}$.

Step 2 Generate x' from some proposal distribution $q(x, x')$ and calculate the acceptance probability $\alpha(x, x')$ of moving from x to x':

$$\alpha(x, x') = \min\left\{1, \frac{\pi(x')\, q(x', x)}{\pi(x)\, q(x, x')}\right\}.$$

Step 3 Draw $u \sim U(0, 1)$. If $u < \alpha(x, x')$, then accept the move and set $X_t = x'$; else, remain in the current state and set $X_t = x$.

This algorithm defines the transition kernel

$$P(x, \mathrm{d}x') = p(x, x')\mu(\mathrm{d}x') + r(x)\delta_x(\mathrm{d}x')$$

with

$$p(x, x') = \begin{cases} q(x, x')\alpha(x, x') & \text{if } x \neq x', \\ 0 & \text{if } x = x', \end{cases}$$

and

$$r(x) = 1 - \int q(x, x')\alpha(x, x')\mu(\mathrm{d}x'),$$

and point mass δ_x. Reversibility is satisfied since

$$\pi(x)p(x, x') = \pi(x')p(x', x) \qquad \text{for all } x, x' \in \Omega.$$

Hence, π is the invariant distribution. Irreducibility and aperiodicity need to be checked on a case-by-case basis. A sufficient condition for irreducibility is that, for every $x \in \Omega$, the proposal distribution satisfies $q(x, x') > 0$ for all $x' \in \Omega$. If in addition the set $B = \{x : r(x) > 0\}$, which contains all the current states to which the sampler is able to return by rejection, has positive probability $\pi(B) > 0$ under π, then the chain is aperiodic.

Actually, Hastings gives a more general class of algorithms by allowing the acceptance probability to be of the form

$$\alpha(x, x') = \frac{s(x, x')}{1 + \frac{\pi(x)\, q(x, x')}{\pi(x')\, q(x', x)}}$$

where $s(x, x')$ is symmetric and such that $0 \leq \alpha(x, x') \leq 1$. Tjelmeland and Hegstad (2001) exploit this feature in their mode jumping method, which we will discuss later. However, the former version of the acceptance probability,

$\alpha(x, x') = \min\left\{1, \frac{\pi(x')\, q(x,x')}{\pi(x)\, q(x',x)}\right\}$, is in general preferred because it minimises the integrated autocorrelation time $\tau(h)$ and thus the asymptotic variance of the MCMC estimator $\bar{h}_N$ for a given proposal distribution $q(x, x')$ [as shown in Peskun (1973) for finite state space and in Tierney (1998) for general state space].

In the Metropolis-Hastings algorithm, we are (relatively) free in choosing a proposal distribution, as long as we can sample from it directly and as long as irreducibility and aperiodicity of the method are ensured. Usually, the proposal distribution is some standard distribution conditional on the current state, and the variance of this distribution is chosen by trial and error: variances that are too small hinder the sampler in mode jumping, variances that are too large hinder the sampler in exploring a single mode. To compromise, sometimes a mixture of standard distributions is employed to ensure good mixing within and between modes; however, the more common way of incorporating different proposal distributions is to combine different kernels by applying them either in a fixed or in a random order (see Section 2.6.2).

If the proposal distribution is symmetric, then the proposal distributions cancel in the acceptance ratio so that $\alpha(x, x') = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}$. This gives the original Metropolis algorithm (Metropolis, Rosenbluth, Rosenbluth, Teller and Teller 1953). For Metropolis algorithms, it is common to choose a random walk proposal, i.e. $x' = x + \varepsilon$ where ε is some random increment. If the target distribution is continuous, we often specify $\varepsilon \sim N_d(0, \sigma^2 I_d)$ (where I_d denotes the identity matrix), or equivalently $q(x, x') \sim N_d(x, \sigma^2 I_d)$. If in addition the target distribution is unimodal and can be written as $\pi(x) = \prod_{i=1}^{d} f(x_i)$, then Roberts and Rosenthal (2001) show that the scaling σ that yields an acceptance rate of 23.4% is most efficient (where efficiency is measured by the inverse autocorrelation time); they also find that any acceptance rate between 10% and 40% is highly efficient so that it is not worth spending much effort in tuning the chain. Unfortunately, these results do not hold if the target is multimodal.

2.6.2 Combining MCMC kernels and Gibbs sampling

Combinations of MCMC kernels are often used for component-wise or block-wise updating. They also allow incorporating different step sizes into the MCMC sampler.

In general, if there are different reversible transition kernels P_i, $i = 1, \ldots, d$, all of which have the same invariant distribution π, then arranging them into a reversible fixed order, e.g. $P = P_1 P_2 \cdots P_{d-1} P_d \cdot P_d P_{d-1} \cdots P_2 P_1$, or into a random order, e.g. $P = \frac{1}{d} \sum_{i=1}^{d} P_i$, yields a new transition kernel P, which is reversible and has invariant distribution π (Besag, Green, Higdon and Mengersen 1995). This is useful if we want to vary the proposal mechanism of the sampler. We can then design one transition kernel, say P_1, for jumping between modes and another, say P_2, for exploring the current mode, and apply them in random order. We also combine kernels when updating a d-dimensional random variable component-wise. The ith kernel P_i is then a reversible transition kernel with equilibrium distribution $\pi(x_i|x_{-i})$, where x_i denotes the ith component and $x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$ the block of remaining components. If q denotes the proposal distribution for this component, we move from x_i to x_i' with probability

$$\alpha(x_i, x_i') = \min \left\{ 1, \frac{\pi(x_i'|x_{-i})\, q(x_i, x_i')}{\pi(x_i|x_{-i})\, q(x_i', x_i)} \right\}.$$

One example of component-wise updating is the Gibbs sampler (Geman and Geman 1984). It draws directly from the conditional distribution to update x_i, i.e. $q(x_i, x_i') = \pi(x_i'|x_{-i})$, and thus always accepts the proposal:

$$\begin{aligned} \alpha(x_i, x_i') &= \min \left\{ 1, \frac{\pi(x_i'|x_{-i})\, q(x_i, x_i')}{\pi(x_i|x_{-i})\, q(x_i', x_i)} \right\} \\ &= \min \left\{ 1, \frac{\pi(x_i'|x_{-i})\, \pi(x_i|x_{-i})}{\pi(x_i|x_{-i})\, \pi(x_i'|x_{-i})} \right\} \\ &= 1. \end{aligned}$$

Geman and Geman (1984) show that also non-reversible sequences of Gibbs kernels P_i, e.g. $P = P_1 P_2 \cdots P_d$, have equilibrium distribution π.

2.7 Convergence diagnostics and perfect sampling

In practice, it is virtually impossible to establish theoretical convergence or mixing rates of the chain. In a few cases, it is possible to assess convergence to equilibrium by coupling several Markov chains (e.g. Propp and Wilson 1996, Johnson (1996,1998), Murdoch and Green 1998, Corcoran and Tweedie 2002, Brooks, Fan and Rosenthal 2006). If the coupled chains coalesce, convergence has been reached. Some of these methods produce a perfect sample (or exact

sample) from the equilibrium distribution π after coalescence. However, these methods can be cumbersome, and they require either starting chains from all possible starting points or a partial ordering of the state space that reduces convergence to the coalescence of a "top" and a "bottom" process.

Apart from the convergence to equilibrium, there is also the convergence of the sample path average $\bar{h}_N = \frac{1}{N}\sum_{i=1}^{N} h(X_i)$ to the expectation $\mathbb{E}_\pi[h(X)]$. For both kinds of convergence, empirical convergence diagnostics have been proposed [for a review, see Cowles and Carlin (1996), Brooks and Roberts (1998) or Robert and Casella (1999, Chapter 7)]. These diagnostics compare either the behaviour of several independent chains run in parallel or the changes of behaviour within a single chain. As long as the behaviour is sufficiently different, the diagnostic correctly signals the lack of convergence. But if no lack of convergence is detected, it is fallacious to assume that convergence has taken place. It may be that the single chain or all the multiple chains are mixing slowly and remain in the same region of the sample space for a very long time. In conclusion, the convergence diagnostic is only as good as the mixing of the MCMC method.

2.8 Burn-in period

As the Markov chain needs to converge, it is common to discard the first iterations of the chain, which are also called the burn-in period. There are two understandings of the required length of burn-in. A minority view is that the burn-in should be equal to the time of convergence to the equilibrium distribution (in the sense that the chain should have forgotten its starting point) and to use convergence diagnostics to determine this period (e.g. Gelman and Rubin 1992, Brooks and Roberts 1998). The majority however follows Geyer (1991, 1992), who points out that, in theory, no burn-in is needed since convergence takes place independently of the distribution of the initial state, but that, in practice, it is sensible to throw away the tail behaviour in the beginning of the chain, namely the iterations it needs to find modal area, because this reduces the bias of the estimator. If the chain settles down in the modal area, it is also common to say that the chain has "converged to equilibrium", although, strictly speaking, the chain has not completely forgotten its starting point because it usually climbs the mode nearest to its starting point. Some authors propose to draw the initial state from an

approximation of the target to shorten the burn-in period, but actually any reasonable starting distribution (e.g. the prior distribution in a Bayesian problem) will often do.

The most important point is that the quality of the sample is much more influenced by the mixing of the chain than by the burn-in: even if the chain has "converged to equilibrium", it might still mix too slowly to be of any use for estimation (Geyer 1992).

2.9 Number of Markov chains

In the beginning, there was much discussion on whether it would be better to base the MCMC estimation on a single chain or on multiple chains (see Geyer (1991, 1992), Gelman and Rubin 1992, Besag and Green 1993 and Tierney 1994). Multiple chains seem to safeguard against multimodality as their starting points can be scattered around the sample space in the hope that all modes will then be found. Again, this can be deceptive: it is essential that the MCMC method mixes well between modes. There is no point in combining chains each of which is stuck in a different mode since their combination will not truly represent the weight of the modes. Again, the most important thing is that the method mixes well within and between modes, and then one sufficiently long run contains all the information. Even if multiple chains are run, it is not clear how to combine them. Gelfand and Smith (1990) suggest to start m chains independently, run them for n iterations (n large), and then to take the very last state of each chain to obtain m hopefully independent samples. But Tierney (1994) warns that it is hard to tell whether n is large enough; Geyer (1991) even dismisses this method as invalid since it requires both m and n to go to infinity. A better approach is to run m independent chains each for n iterations (after burn-in), and then to estimate the expectation $\mathbb{E}_\pi[h(X)]$ by the pooled mean $\frac{1}{mn}\sum_{i=1}^{m}\sum_{t=1}^{n} h(X_{it})$; but again this requires n to be of the same length as a sufficient long single run, in which case no additional information is gained by the multiple run at the m-fold cost (which also includes the cost of discarding m burn-in periods rather than just one). In conclusion, multiple runs are only useful to discover multimodality. This information should be used to design MCMC methods that are mixing well. Once such a method is found, a single long run suffices for inference.

Chapter 3

Mode Jumping Methods in Fixed Dimension: a Review

3.1 Introduction

The mode jumping problem arises from the reluctance of MCMC to give up a mode for a low-probability area. MCMC is virtually unable to move from one mode to another by taking several steps through low-probability areas. The only way in which it can reach another mode is a direct jump (in one step). In standard MCMC, such a jump is difficult to achieve because new proposals are drawn "blindly" from the surrounding area of the current mode and thus are much more likely to hit low-probability area than another mode. This problem occurs in particular if modes are spiky (i.e. with much mass concentrated on a small spot) or if the dimension of the problem is high. If the chain is in addition updated component-wise, then the limited number of proposal directions restricts the sampler so that some modes may be unattainable.

We can tackle the mode jumping problem in two complementary ways. One way is choosing (or designing) a mode jumping method, the other is monitoring and, if necessary, improving (or changing) the method. For this, it is helpful to find out more about the nature of the sampling problem (in particular about the number and location of modes) by preliminary MCMC runs or by preliminary mode searches (Section 3.2). It is also useful to learn about the existing mode jumping approaches. This review will focus on the key ideas behind the methods. The first of the five key ideas is to learn from the history of the chain (or from other chains) and to propose more states from previously visited areas so that the chain may return to modes it has

seen so far (Section 3.3.1). The second idea ("slice sampling") defines cross-sections ("slices") through the modes and moves from mode to mode along these cross-sections (Section 3.3.2). The third approach takes excursions over a distribution defined on an unconstrained state space or on a higher-dimensional state space in which mixing between modes is possible (Section 3.3.3). The fourth idea involves deterministic mode searches so that proposals are not anymore drawn "blindly" from the neighbourhood of the current state, but directly from one of the modes (Section 3.3.4). The last approach incorporates tempered versions of the distribution which feature less definite modes with larger basins of attraction so that the sampler can easily leave the current mode and move to another one (Section 3.3.5). There is a limit to what can be said about how the methods perform in comparison to each other because there are not many comparative studies available and because it is difficult to assess the implementation effort, computational cost and efficacy of the methods without having implemented them.

3.2 Preliminary mode searches

In complex problems, it is recommended to learn about the target distribution through preliminary MCMC runs and mode searches, which may be conducted by starting the sampler from several starting points scattered around the sample space. The information gained through these preliminary runs may help designing a well-mixing MCMC method and assessing the convergence of the sampler.

An unrealistic goal of such mode searches is to find all the modes of the distribution and to approximate these modes by standard distributions. In this case, we could use these approximating standard distributions as independent proposal distributions in the MCMC algorithm so that a proposal state is drawn from one of the known modes independently of the current state. Such a chain would mix very well between modes. Recall however that MCMC is a last resort which should only be used if simpler methods (such as approximations based on exhaustive mode searches) are not feasible. In a way, we have already ruled out the possibility of approximating the target by standard distributions when choosing MCMC.

The information from preliminary runs may be incomplete because there is

always the possibility that modes are missed out. The information is however valuable because it can be used to assess and improve the current MCMC method. It may give us a vague idea of the important region of the distribution so that the proposal distributions can be chosen such that the sampler can move within this region. In Bayesian problems, for instance, the prior distribution is often vague and remains very dispersed compared to the posterior distribution. In this case, independent proposals from the prior distribution may help the sampler to move around the sample space although a sampler solely based on independent proposals may be inefficient. The information about modes also allows us to check the mixing of the chain, even if the information is incomplete. If the chain cannot jump between known modes, it is definitely not mixing well and needs improvement. If a method that is able to discover new modes mixes well between the known modes, we can be fairly confident that it can also visit any unknown modes within the known region. Good mixing within this region does not safeguard against missing modes which lie far away from this region. In practice, this risk seems negligible because we usually have a vague idea of the main support of the target distribution in the sense that we would not expect any modes outside this support. If we combine this information with the information from the preliminary runs, we should be able to construct a sampler which mixes well within the relevant part of the sample space.

3.3 Mode jumping methods

3.3.1 Learning from the past and learning from other chains

One type of mode jumping method tries to learn about the target distribution while sampling and to incorporate this information into the sampling process. Such methods are called adaptive methods. They either learn from a population of parallel chains (all under the same target) or from the history of a single chain. When updating a population of parallel chains, a new generation is created based on information available in the parent generation. Ideally, the parent generation occupies all the modes of the distribution. Mode swaps between a single parent and its off-spring are also possible because each off-spring depends on the entire parent generation (and not only on the parent it replaces) [see for example Gilks, Roberts and George (1994), Braak (2006) and Liu, Liang and Wong (2000) for different variants of this idea].

Learning from the past of a single chain is more difficult to implement because the next state of a Markov chain can only depend on the current state, but not on any of the past states. However, a Markov chain may be stopped at regeneration times, i.e. times when the chain has forgotten its starting point, and its transition kernel may be modified. After modification, the Markov chain can continue its sampling under the adapted transition kernel and both the old and the new samples can be used for estimation. Gilks, Roberts and Sahu (1998) provide a theoretical framework for adapting Markov transition kernels at regeneration times. If the state space is discrete, regeneration occurs whenever the chain passes through a nominated state. If the state space is continuous, it is much more difficult to determine regeneration times so that this idea can hardly be applied in practice.

In general, one should be careful in designing adaptive algorithms as some of them may sample from the wrong distribution (Atchadé and Rosenthal 2005) or converge to suboptimal values [see for example Jennison and Sheehan (1995) and Franconi and Jennison (1997)]. Another danger in population Monte Carlo is that the entire population is trapped in one area of the sample space so that other areas can hardly be reached. To overcome this problem, some population MCMC methods incorporate "tempered" versions of the target [see Section 3.3.5, "Metropolis-coupled MCMC (parallel tempering)", for a brief explanation and literature review].

3.3.2 Slice sampling

Another school of thought tackles the mode jumping problem by introducing auxiliary variables, over which previously separated modes can be reached. A classical auxiliary variable method is slice sampling, which can directly move between modes, at least in theory. In its simplest version, the slice sampler samples from $\pi(x) \propto f(x)$ on Ω by sampling from the joint distribution

$$p(x,u) \propto \begin{cases} 1 & \text{if } 0 < u < f(x), \\ 0 & \text{otherwise,} \end{cases}$$

on $\Omega \times \mathbb{R}$ and keeping only the x samples. This works because $p(x,u)$ has marginal distribution $p(x) \propto f(x)$:

$$p(x) = \int p(x,u)\,\mathrm{d}u \propto \int_0^{f(x)} \mathrm{d}u = f(x).$$

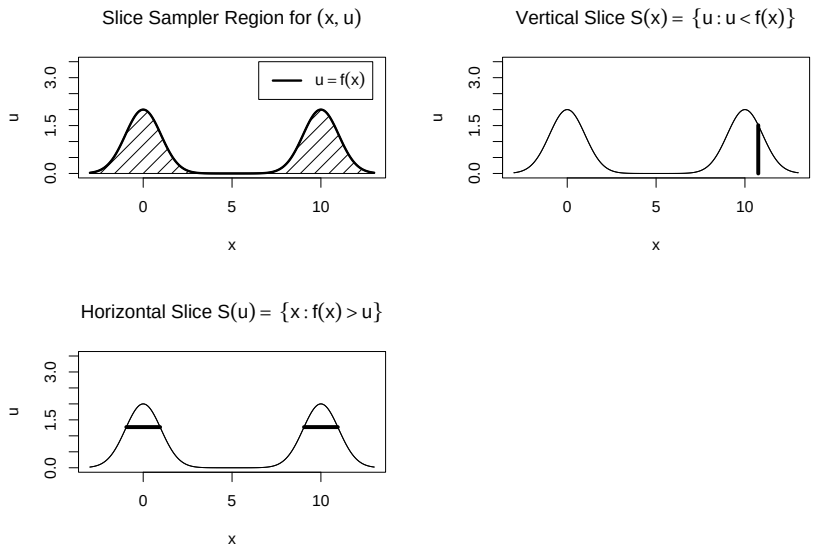

Figure 3-1: The support of the joint distribution $p(x, u)$ is shown as shaded area *(top left)*. When sampling from the joint distribution, x and u are updated alternately. Given x, a new u is chosen uniformly from the vertical slice $S(x) = \{u : 0 < u < f(x)\}$ *(top right)*. Given u, a new x is chosen uniformly from the horizontal slice $S(u) = \{x : f(x) > u\}$ *(bottom left)*.

The slice sampler updates u and x alternately (see Figure 3-1): first a new u is drawn uniformly from the vertical slice $S(x) = \{u : 0 < u < f(x)\}$ between zero and the current $f(x)$ value, i.e. $u \sim U(0, f(x))$; then a new x is drawn uniformly from the horizontal slice $S(u)$ through u, namely $S(u) = \{x : f(x) > u\}$, which contains all the states with higher density values than the current one. As the horizontal slice forms a cross-section through the modes of the target distribution $\pi(x)$, mode jumps are possible. Slice sampling originates in the Swendsen-Wang algorithm designed for image analysis, and has been developed since (Swendsen and Wang 1987, Edwards and Sokal 1988, Besag and Green 1993, Higdon 1998, Damien, Wakefield and Walker 1999, Mira, Møller and Roberts 2001, Roberts and Rosenthal 2002). Overall, slice sampling has very good convergence properties (see for example Roberts and Rosenthal 1999, Mira and Tierney 2002). But often, slice sampling is impracticable because the slices cannot be determined, for example if the locations of modes are unknown. Therefore, Neal (2003) suggests searching for parts of the slice within which the sampler can move in the neighbourhood of the current state; but then again, modes far off may be missed out.

3.3.3 Excursions over a different model

In some applications, e.g. in Mendelian genetics, there are constraints such that large areas of the product sample space have zero probability under the target distribution so that moving between two disconnected parts of the sample space is difficult. These parts may however be reached by excursions over an unconstrained distribution (Hurn, Rue and Sheehan 1999): if the Markov chain, which samples from the constrained distribution, is currently in x, then a secondary chain is started in x to sample from the unconstrained distribution until a state x' is reached which is also feasible under the constrained distribution. This state x' is the proposal for the Markov chain sampling from the constrained distribution and thus either accepted or rejected.

In other applications, e.g. in Bayesian mixture modelling, excursions over higher-dimensional models can help the sampler in moving between the modes of a particular fixed-dimensional model (Richardson and Green 1997). This works because the larger model contains variables which are not necessary to explain the data. As a result, the joint distribution of higher dimension is much more diffuse than the one of lower dimension so that transitions between modes are not a problem in high dimension. This approach requires sampling from variable-dimensional distributions, which we will discuss in Chapter 8.

3.3.4 Mode jumping via local optimisation

Mode jumping via local optimisation constructs a proposal mechanism which takes first a large random step into possibly low-probability area and then finds the closest mode by a deterministic mode search, from which a proposal is chosen (Tjelmeland and Hegstad 2001, Tjelmeland and Eidsvik 2004). This method will be described in detail in Chapter 4, where it is also tested in comparison to another mode jumping method "tempered transitions". Although the idea is very neat, the implementation is difficult because deterministic mode searches suffer from numerical instability in low-probability areas if the density in these areas is computationally equal to zero.

3.3.5 Tempering methods

Basic idea

The basic idea behind "tempering" is to control the shape of the target density by a "temperature" parameter. If tempering is used for mode swapping, the shape is "softened" so that modes "melt" together and transitions between modes become feasible. The converse is also possible: temperatures may exacerbate the shape, which helps to find the modes of a distribution. Indeed, the idea of tempering originates in the stochastic optimisation method "simulated annealing" (Kirkpatrick, Gelatt and Vecchi 1983, Geman and Geman 1984), which will be described in Section 5.4.2.

The classic way to temper the target distribution $\pi(x)$ is to define its tempered version by $\pi_\beta(x) \propto [\pi(x)]^\beta$ where β is called the "inverse temperature". It is also possible to temper only one part of the distribution. In Bayesian statistics, we may for example only temper the prior or the likelihood contribution of the target posterior distribution. Tempering methods also work if the tempered distributions are replaced by another type of auxiliary distribution, for example by unconstrained versions of the target distribution, as long as the resulting algorithm satisfies irreducibility and aperiodicity.

In this section, we will assume for simplicity that the tempered distributions have the classical form $\pi_\beta(x) \propto [\pi(x)]^\beta$. If the inverse temperature is equal to one ($\beta = 1$), the density takes its original shape. The smaller the inverse temperature becomes ($\beta \to 0$), the more the modes spread out so that the density "flattens". For mode jumping, it already helps if the modes are loosely connected. When applying tempering in MCMC, the inverse temperatures smaller than one at which modes merge together are "hot" inverse temperatures and, similarly, the distributions they define are "hot" distributions, while inverse temperatures close to one are "cold" inverse temperatures and the corresponding distributions are "cold" distributions. When applying tempering in stochastic optimisation, the notion of "cold" and "hot" changes. In stochastic optimisation, the "cold" temperatures ($\beta \gg 1$) are those, at which the mass of the distribution is contracted at the maxima of the density.

The simplest tempering idea in MCMC is to use a tempered version of the target as auxiliary distribution for importance sampling based on MCMC samples (Jennison 1993). A more elaborate approach defines a sequence

of tempered importance distributions which slowly approaches the target distribution. If this sequence is well chosen, this yields a more efficient importance sampler than the simple importance sampler (Neal 2001).

In the following, three similar tempering methods, namely simulated tempering, Metropolis-coupled MCMC (parallel tempering) and tempered transitions, are discussed in more detail. This discussion will show the advantages and disadvantages of each method. In general, tempering methods perform very well although they require a lot of tuning.

Simulated tempering

Simulated tempering samples from the target distribution $\pi(x)$ indirectly (Marinari and Parisi 1992, Geyer and Thompson 1995). It is set up to sample from another distribution

$$p(x,i) \propto \psi(i)\ [\pi(x)]^{\beta_i}\,, \qquad x \in \Omega,\ i \in \{0,1,\ldots,n\}\,,$$

where $\psi(i)$ is an auxiliary distribution ("pseudo-prior") and $\{\beta_i\}_{i=0}^n$ is a set of inverse temperatures such that $0 < \beta_n < \beta_{n-1} < \cdots < \beta_0 = 1$. Both the inverse temperatures and the pseudo-prior need to be chosen in advance and require some tuning, which we will discuss after describing the algorithm. By the choice of inverse temperatures, the conditional distribution $p(x|i=0)$ is identical to the target distribution $\pi(x)$ so that we obtain a sample from the target by keeping only the samples from the joint distribution $p(x,i)$ that are generated at the target temperature β_0 (i.e. when $i=0$).

In simulated tempering, the variable x and the auxiliary variable i are updated alternately. When updating x (given i), an MCMC step with respect to the equilibrium distribution

$$\begin{aligned} p(x|i) &\propto \psi(i)\ [\pi(x)]^{\beta_i} \\ &\propto [\pi(x)]^{\beta_i} \end{aligned}$$

is carried out. When updating i (given x), the proposal is set to $i' = i \pm 1$ with probability $q(i,i+1) = q(i,i-1) = \frac{1}{2}$ if $1 < i < n$ and with probability $q(1,2) = q(n,n-1) = 1$ otherwise. This proposal is then accepted with the Metropolis-Hastings acceptance probability

$$\alpha(i,i') = \min\left\{1, \frac{\psi(i')\ [\pi(x)]^{\beta_{i'}}\ q(i',i)}{\psi(i)\ [\pi(x)]^{\beta_i}\ q(i,i')}\right\}.$$

The advantage of this method is that mode swaps can take place at high temperatures. The disadvantage is that the sampler wanders erratically up and down the temperature ladder so that it may take a long time until the distribution $p(x|i=0)$ of interest is reached. The length of these excursions is considered the cost of the method. Geyer and Thompson (1995) give a rough guidance for the expected cost by comparing updating i to simulating a random walk on $\{0, 1, \ldots, n\}$ which moves from the current state to one of the adjacent states with probability $\frac{p}{2}$. This means that the random walk stays in the current state with probability $(1-p)$ unless the current state is one of the endpoints in which case it remains there with probability $\left(1-\frac{p}{2}\right)$. The expected cost of going from $i=0$ to $i=n$ is then $\frac{n\,(n+1)}{p}$. If we assume that simulated tempering behaves similarly and comes at cost $n\,(n+1)/(\text{acceptance rate})$, then it is only worth doubling the number of inverse temperatures n if the acceptance rate of proposing a new i multiplies by more than a factor of four. If the acceptance rate is already above 0.25, then doubling the number of temperatures will always be inefficient. Since this random walk does not exactly model the behaviour of simulated tempering, acceptance rates of 0.2 to 0.4 are considered reasonable in simulated tempering. Apart from the number of temperatures, the spacing of temperatures matters too. If two adjacent temperatures lie too far apart, the corresponding tempered distributions do not match properly in the sense that the current x may be likely under the current tempered distribution, but unlikely under the adjacent distribution. In this case, the acceptance rates will be low because of this mismatch. If the spacing between temperatures is well chosen, then the chain will be rapidly mixing between the distributions. Ideally, the chain should spend an equal amount of time under each temperature, in other words the probabilities

$$\begin{aligned}\mathbb{P}\{I=i\} &\propto \psi(i)\int_{\Omega}\left[\pi(x)\right]^{\beta_i}\,\mathrm{d}\mu(x)\\ &= \psi(i)\,Z(i)\end{aligned}$$

[where $Z(i)$ is the normalisation constant for the distribution $p(x|i)$] should be equal. This can be achieved by tuning the pseudo-prior such that $\psi(i)=\frac{1}{Z(i)}$. This tuning is also tedious. Some strategies can be found in Geyer and Thompson (1995).

Apart from the mode jumping ability, simulated tempering has the advantage that regeneration times can be incorporated. Suppose, in a Bayesian context, the tempered distribution is of the alternative form $p(x,i) \propto \pi(x)\left[l(x)\right]^{\beta_i}$, where $\pi(x)$ is the prior and $l(x)$ is the likelihood contribution. If we can draw

independent samples from the prior distribution, then setting $\beta_n = 0$ means that we generate independent samples whenever the chain passes through this temperature (through $i = n$) so that regeneration takes place. Regeneration speeds up the mixing of the sampler and can improve estimation.

Usually only the samples from the conditional distribution $p(x|i = 0)$ are retained because these are the samples from the target distribution. Gramacy, Samworth and King (2007) consider this wasteful. They collect all the samples from the conditional distributions $p(x|i)$, $i = 0, \ldots, n$. For a given i, they use the samples from $p(x|i)$ to estimate a certain quantity by importance sampling. They repeat this for every possible i so that they have in total $(n + 1)$ importance estimates of the same quantity. These estimates are then combined in an optimal way so that the pooled estimate has a higher accuracy than each of the original $(n + 1)$ importance estimates.

Metropolis-coupled MCMC (parallel tempering)

Metropolis-coupled MCMC (parallel tempering) employs $(n+1)$ chains run in parallel, but each at a different temperature so that the ith chain, $i = 0, \ldots, n$, samples from the tempered distribution $\pi_{\beta_i}(x) \propto [\pi(x)]^{\beta_i}$ where the inverse temperatures are again chosen such that $0 < \beta_n < \cdots < \beta_0 = 1$ (Geyer 1991). At hot temperatures, the parallel chains mix well between modes. At cold temperatures, mixing between modes is difficult within each chain. The trick of Metropolis-coupled MCMC is to couple hot and cold chains in such a way that the cold chains benefit from the fast mixing of the hot chains: after updating all the chains individually, state swaps between adjacent chains are considered. Mode swaps will take place if adjacent chains currently visit two different modes and the state swap between these chains is accepted. If x_i is the current state of the π_{β_i}-invariant chain and x_j is the state of the π_{β_j}-invariant chain, then a swap is accepted with probability $\alpha = \min\left\{1, \frac{\pi_{\beta_i}(x_j)\pi_{\beta_j}(x_i)}{\pi_{\beta_i}(x_i)\pi_{\beta_j}(x_j)}\right\}$.

The advantage of Metropolis-coupled MCMC over simulated tempering is that it does not require the calculation of normalisation constants. The disadvantage is that storing m chains in parallel is quite costly, and that the mixing between these m chains is not as good as in simulated tempering (Geyer and Thompson 1995).

The idea of parallel tempering finds a further development in the "equi-energy

sampler", in which slice sampling ideas are used for coupling chains run under different temperatures (Kou, Zhou and Wong 2006). The equi-energy sampler uses the fast exploration of the hot chains to gain information about regions ("rings") of equal energy under the target distribution. The energy rings are defined across modes. By jumping between states of equal energy, the sampler is able to move between modes. The disadvantage of the algorithm is that the visited states have to be sorted and stored in such a way that any previous state can be picked at random for the equi-energy jump. This makes the method cumbersome and very expensive.

Another branch of methods arises from combining Metropolis-coupled MCMC and population MCMC (see Section 3.3.1). Both methods have in common that the parallel chains (once under different temperature and once under the same temperature) learn from each other. While Metropolis-coupled MCMC is able to explore the entire space due to the fast mixing of the hot chains, population MCMC can get stuck in one part of the sample space provided that all chains are run under the same temperature. On the other hand, population MCMC uses a greater variety of move types that learn from other chains than the original Metropolis-coupled MCMC algorithm. A branch of methods therefore combines parallel tempering and population MCMC by running each chain (i.e. each member of the population) under a different temperature and allowing not only swap moves (as in Metropolis-coupled MCMC), but also other cross-over moves known from population MCMC to improve the overall mixing (see for example Jasra, Stephens and Holmes 2007a, Goswami and Liu 2007, Liang and Wong 2001).

Another way of combining population MCMC and tempering ideas is used in sequential Monte Carlo. Sequential Monte Carlo generates N weighted samples $\left\{w_0^{(i)}, x_0^{(i)}\right\}_{i=1}^{N}$ (where the weights sum to one, $\sum_{i=1}^{N} w_0^{(i)} = 1$) from the tempered distribution p_{β_0}. It starts with a population of N weighted samples $\left\{w_n^{(i)}, x_n^{(i)}\right\}_{i=1}^{N}$ from the hottest distribution p_{β_n}. Given these samples, it produces the next generation of N samples by MCMC transition kernels and reweights (or resamples) the new population with respect to the next tempered distribution $p_{\beta_{n-1}}$ so that an estimate based on these weighted samples $\left\{w_{n-1}^{(i)}, x_{n-1}^{(i)}\right\}_{i=1}^{N}$ is consistent with respect to $p_{\beta_{n-1}}$. This sequential sampling principle is repeated with respect to the remaining distributions $p_{\beta_{n-2}}, \ldots, p_{\beta_0}$ so that the sequential sampler ends up with a weighted sample

from p_{β_0} giving consistent estimates with respect to the target distribution. For more details, see for example Jasra et al. (2007a), Del Moral, Doucet and Jasra (2006), Eberle and Marinelli (2006), Chopin (2004), Doucet, Godsill and Andrieu (2000), and Liu and Chen (1998).

Tempered transitions

Tempered transitions is a single-chain method which takes excursions over all the tempered auxiliary distributions $\pi_{\beta_i}(x) \propto [\pi(x)]^{\beta_i}$, $0 \leq \beta_n < \cdots < \beta_0 = 1$, in a fixed deterministic order to create a proposal for the target distribution $\pi_{\beta_0}(x)$ (Neal 1996). When working with tempered transitions later, we will use temperature schemes of the form $0 \leq \beta_n < \cdots < \beta_1 = \beta_0$ to ease the presentation of experiments. We will explain this when describing tempered transitions in more detail in Section 4.2. Here it suffices to say that, in tempered transitions, the proposal mechanism for the primary chain which samples from the target distribution starts a secondary chain at the current state x_0. This secondary chain passes through all the auxiliary distributions first in ascending order (so that the modes flatten), then in descending order (so that the modes regain their shape). Since the secondary chain can travel freely through the sample space at least at the hottest temperature, its last state x'_0 may come from a different mode. This state is then accepted as the next state of the primary chain with a probability α that depends on the secondary chain $(x_0, x_1, \ldots, x_n, x'_{n-1}, \ldots, x'_1, x'_0)$ and the inverse temperatures $\{\beta_i\}_{i=0}^n$:

$$\alpha(x_0, x_1, \ldots, x_n, x'_{n-1}, \ldots, x'_1, x'_0) = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right] \left[\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\}.$$

If the proposal is not accepted, the primary chain will remain in the current state x_0 for another iteration.

As with all tempering methods, tempered transitions requires tuning the number and spacing of temperatures. The advantage over other tempering methods is that it does not require any normalisation constants as these would cancel anyway in the acceptance probability. Furthermore, tempered transitions is based on a single chain so that it is cheaper in storage than parallel tempering. In comparison to simulated tempering, the cost of an excursion is fixed. It is proportional to $2n$. At first glance, the cost seems to be lower than in simulated tempering. However, tempered transitions requires a higher number of temperatures than simulated tempering to obtain good mixing so that, in the end, the computational effort is similar (Neal 1996). It

seems that tempered transitions is the tempering method which is easiest to implement.

Another way of looking at tempered transitions is to think of the proposal mechanism as a mode searching method. In the proposal mechanism, the temperatures are used in such a way that the basins of attraction of the modes first expand and then contract so that the proposal mechanism can easily move away from the current mode and find a new mode for the final proposal. In this light, tempered transitions is comparable to mode jumping via local optimisation, which incorporates deterministic mode searches. We will test both methods in the next chapter.

Chapter 4

Tempered Transitions versus Mode Jumping via Local Optimisation

4.1 Introduction

Tempered transitions and mode jumping via local optimisation are both promising mode jumping methods. They are reported to perform well when sampling from a mixture of normal distributions, which is a notorious hard sampling problem. It is not clear which method to prefer because they have both advantages and disadvantages: tempered transitions (Section 4.2) is relatively simple to code, but requires possibly tedious tuning of parameters, while mode jumping via local optimisation (Section 4.3) hardly requires any tuning, but is unwieldy in its implementation. Similarly, we do not know which method performs better. To get a feeling for the implementation difficulties and the quality of performance, we will test both methods on a toy problem (Section 4.4).

4.2 Tempered transitions

4.2.1 Tempered distributions

As discussed in Section 3.3.5, the general idea behind tempering methods such as tempered transitions is to enable good mixing, in particular between modes, by incorporating over-dispersed versions of the target distribution into the sampling mechanism. Usually, these over-dispersed auxiliary distributions are tempered versions of the target, but also other types of auxiliary distributions are possible, for example unconstrained versions of the target distribution if

the target distribution is constrained.

We will here focus on a very common class of tempered distributions. This class assumes that the target distribution can be expressed by

$$p(x) \propto \pi(x) \exp\left[-\beta_0\, h(x)\right],$$

where $h(x)$ is called the energy function and the parameter β_0 the target inverse temperature. Since we can write any positive function $f(x)$ in exponential form $f(x) = \exp\left[-\beta_0\, h(x)\right]$ by setting $h(x) := -\frac{1}{\beta_0} \log\left[f(x)\right]$ where usually $\beta_0 = 1$, this class covers a wide range of applications. The part of the target distribution that is expressed in exponential form is the part which we will temper. The tempered distributions are defined by

$$p_{\beta_i}(x) \propto \pi(x) \exp\left[-\beta_i\, h(x)\right], \qquad i = 0, 1, \ldots, n,$$

where β_i, $i = 0, 1, \ldots, n$, are the inverse temperatures. The wide applicability is not the only reason for choosing this particular way of tempering. This form also helps us in reducing the cost of tempered transitions (Chapters 5 to 7). Furthermore, it gives us the choice of tempering the distribution either fully (by setting $\pi(x) \propto 1$ for all x) or part-wise (by choosing a non-constant $\pi(x)$). This choice is an advantage because tempering the entire distribution may define improper distributions, which make the algorithm invalid. Improper tempering can be avoided by tempering only one part of the distribution. It is recommended to pick the part that causes the multimodality. Again, we have to make sure that the tempered distribution is proper. This is easy in the common case that the target distribution is a posterior distribution whose multimodality is caused by the likelihood function. In this case, it is sensible to temper the likelihood part only, while leaving the prior part unchanged. As long as the posterior and the prior distribution are proper, this definition always gives a proper tempered distribution so that no further checking is necessary (Section 7.5). An example for multimodality caused by the likelihood function is the "label-switching" problem in Bayesian mixture modelling. In mixture modelling, permuting the components of a particular model gives an equivalent model which has the same likelihood. In consequence, each permutation has its own mode (see Section 7.2 for further discussion).

The likelihood function is not always the problem. It may also happen that the prior distribution is multimodal, while the likelihood is not. In this case, the prior distribution should be expressed by $\exp\left[-\beta_0\, h(x)\right]$ and then tempered as

above. An example of a multimodal prior is the Ising model used in image analysis. The a-priori model assumes that the pixels of an image can only take two values 0 or 1 (e.g. "black" or "white") and that a particular image (e.g. a white subject on a black background) is equivalent to the corresponding negative image (e.g. a black subject on a white background). The prior distribution has thus two mirroring modes between which standard MCMC often fails to switch.

Tempering methods do not only improve the mixing between modes, but also within modes. For example, when a mode has a heavy tail, standard MCMC may spend long periods of time in the tail, in particular if the tail is very long or flat, so that the overall mixing is very slow. Tempering a heavy-tailed distribution helps the sampler find back to the peak of the mode (see Figure 4-1). Mixing problems also occur if a mode is long and thin: if standard MCMC proposes large steps, it will hardly hit high-probability areas; if it proposes small steps, it will need a very long time to move from one end of the mode to another. In this case, tempering helps because it increases the mode width (see Figure 4-2).

4.2.2 Algorithm

Let us describe tempered transitions (Neal 1996) with respect to sampling from a target distribution of the form

$$p(x) \propto \pi(x) \exp\left[-\beta_0\, h(x)\right]$$

by excursions over the tempered distributions

$$p_{\beta_i}(x) \propto \pi(x) \exp\left[-\beta_i\, h(x)\right], \qquad i = 0, 1, \ldots, n.$$

The inverse temperatures are chosen such that $0 \leq \beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$. We will see later that this ordering is important for the efficiency of the method (Section 5.3.2). Note that $p_{\beta_0}(x)$ is identical to the target distribution. For efficiency, the smallest ("hottest") inverse temperature $\beta_{\min}$ should not be chosen much smaller than necessary to allow mixing between modes (Section 5.2). The efficiency also depends on the number and the spacing of the inverse temperatures. We will therefore discuss optimal choices in Chapters 5 and 6. In this chapter, we will use the default choice of spacing the inverse temperatures geometrically so that we only have to tune the hottest temperature and the number of temperatures between the hottest and the target temperature, which

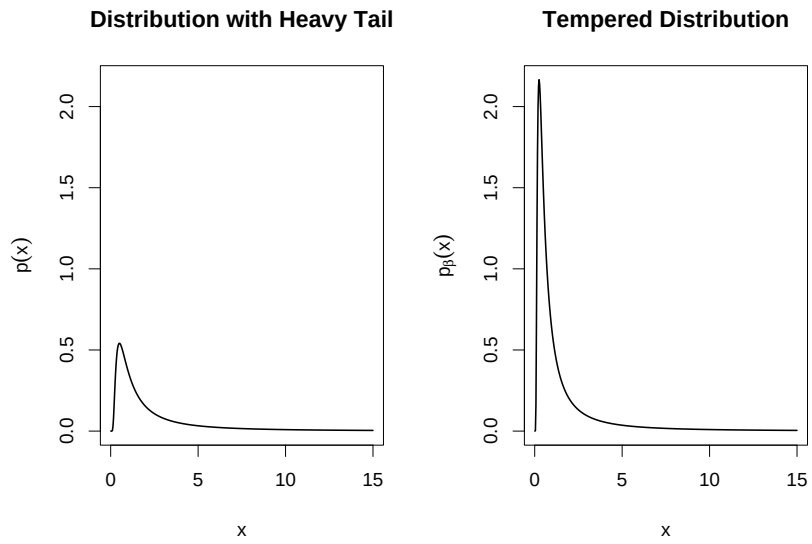

Figure 4-1: A distribution with heavy tail and its tempered version.

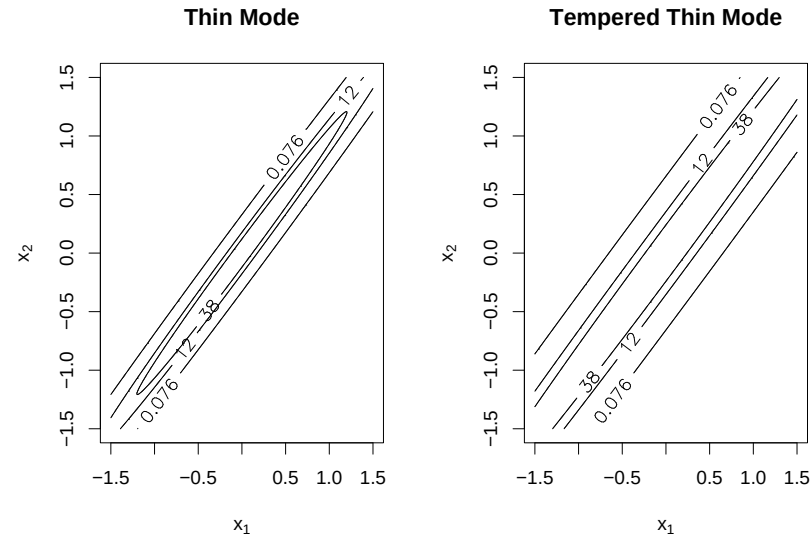

Figure 4-2: A contour plot of a distribution with a thin mode and a contour plot of its tempered version.

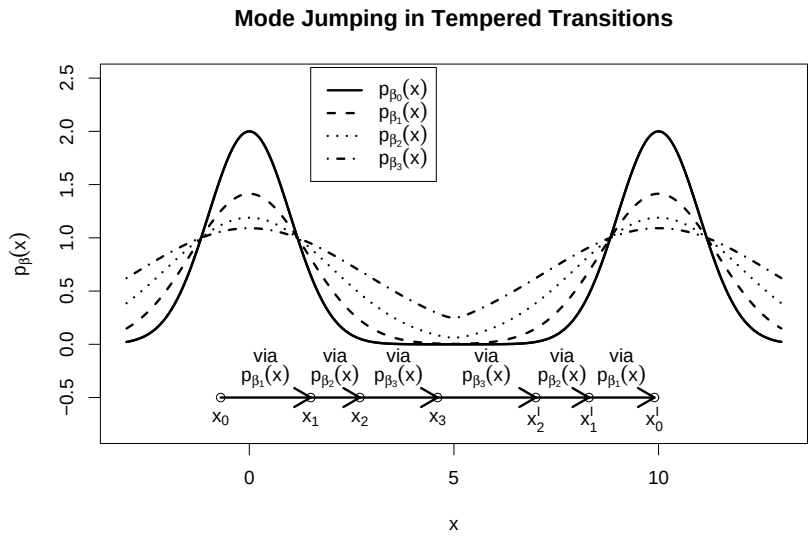

Figure 4-3: A particular tempered transitions path to propose a mode swap for p_{β_0} via the auxiliary distributions $p_{\beta_1}, p_{\beta_2}, p_{\beta_3}$.

can be done by trial and error.

The method of tempered transitions generates a new proposal for the current state using a secondary chain which passes through all the auxiliary distributions $\{p_{\beta_i}\}$ first in ascending order ("heating-up" for enabling mode jumping) and then in descending order ("cooling-down" for coming back to the target distribution p_{β_0}). Figure 4-3 shows an idealised mode swapping path generated by this secondary chain. Before we can define the secondary chain, we need MCMC transition kernels for the auxiliary distributions p_{β_i}, $i = 1, \ldots, n$. In the original version of the algorithm, different Markov transition kernels may be used at the same temperature level β_i depending on the direction of the movement: the transition $\hat{T}_{\beta_i}(x, x')$ for heating-up and the transition $\check{T}_{\beta_i}(x, x')$ for cooling-down, as long as the pair of them satisfies $p_{\beta_i}(x)\,\hat{T}_{\beta_i}(x, x') = p_{\beta_i}(x')\,\check{T}_{\beta_i}(x', x)$ for all x, x'. This distinction gives us the possibility to carry out several sub-transition kernels S_j, $j = 1, \ldots, k$, at the same temperature level β_i at a relatively low cost provided that all the sub-transition kernels satisfy detailed balance with respect to p_{β_i}. This can be done by setting $\hat{T}_{\beta_i} := S_1 \cdots S_k$ and $\check{T}_{\beta_i} := S_k \cdots S_1$, which halves the cost of an algorithm using the same reversible transition kernel $\tilde{T}_{\beta_i} = S_1 \cdots S_k \cdot S_k \cdots S_1$ for both $\hat{T}_{\beta_i}$ and $\check{T}_{\beta_i}$. Defining different transition kernels for the heating-up and the cooling-down process is not the only way of reducing the cost. In the

above example, we can also half the cost by applying each of the sub-transitions S_j, $j = 1, \ldots, k$, only once but in random order independent of whether we are moving up or down the temperature "ladder". In the following, we will only work with random order transition kernels T_{β_i}, $i = 1, \ldots, n$. We will therefore simplify the original algorithm by setting $\hat{T}_{\beta_i} = T_{\beta_i}$ and $\check{T}_{\beta_i} = T_{\beta_i}$, $i = 1, \ldots, n$, where every T_{β_i} satisfies the detailed balance condition

$$p_{\beta_i}(x)\, T_{\beta_i}(x, x') = p_{\beta_i}(x')\, T_{\beta_i}(x', x) \qquad \forall\, x, x' \in \Omega.$$

In this simplified version, the tempered transitions algorithm proceeds at each iteration $t = 1, 2, \ldots, N-1$ as follows:

Algorithm 4.1:

Step 1 Set $x_0 = X_{t-1}$.

Step 2 Draw x_0' as follows:

Generate x_1 from x_0 using the MCMC transition kernel T_{β_1}.
Generate x_2 from x_1 using the MCMC transition kernel T_{β_2}.
$\vdots$
Generate x_n from x_{n-1} using the MCMC transition kernel T_{β_n}.
Generate x'_{n-1} from x_n using the MCMC transition kernel T_{β_n}.
$\vdots$
Generate x'_1 from x'_2 using the MCMC transition kernel T_{β_2}.
Generate x'_0 from x'_1 using the MCMC transition kernel T_{β_1}.

Step 3 Calculate the acceptance probability of moving from x_0 to x_0', which depends on the particular path of the auxiliary transitions:

$$\begin{aligned} &\alpha(x_0, x_1, \ldots, x_n, x'_{n-1}, \ldots, x'_1, x'_0) \\ &\quad = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right] \left[\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\}. \end{aligned} \tag{4.1}$$

Step 4 Draw $u \sim U(0,1)$. If $u < \alpha(x_0, x_1, \ldots, x_n, x'_{n-1}, \ldots, x'_1, x'_0)$, then accept the move and set $X_t = x_0'$; else, remain in the current state and set $X_t = x_0$.

As many steps have to be carried out to generate a proposal state, tempered transitions is an expensive method. We will investigate how to reduce its cost in Chapters 5 to 7. When working with tempered transitions, we deliberately use the temperature scheme

$$\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0 \tag{4.2}$$

instead of the original scheme $\beta_{\min} = \beta_n < \cdots < \beta_1 < \beta_0$ because it allows us to describe experiments in an unambiguous manner. The original definition causes some confusion for there are in total $(n+1)$ distinct temperature levels between the hottest temperature $\beta_{\min}$ and the target temperature β_0, but the secondary chain passes only through n temperatures $\beta_1, \ldots, \beta_n$ so that the computational cost is proportional to n. The problem becomes apparent when we write that we base tempered transitions on 60 geometrically spaced temperatures between $\beta_{\min}$ and β_0. This could mean either that the secondary chain passes through 60 temperatures, in which case $\beta_i = \beta_{\min}^{i/60}$, $i = 0, \ldots, 60$, or that there are in total 60 temperatures of which the secondary chain uses only 59, in which case $\beta_i = \beta_{\min}^{i/59}$, $i = 0, \ldots, 59$. As both interpretations define two different experiments, this confusion should be avoided. The simplest way to bring clarity into the description of experiments is to let n denote the number of distinct temperature levels between $\beta_{\min}$ and β_0 and to let the secondary chain pass through all of them by defining (4.2). In this case, n is also the highest temperature index so that, for example, basing tempered transitions on 60 geometric temperatures unambiguously means setting $\beta_i = \beta_{\min}^{(i-1)/59}$, $i = 1, \ldots, 60$. This simplification implies that the scheme is slightly inefficient due to the duplication $\beta_1 = \beta_0$. However, this inefficiency can be neglected in practice because usually a large number of temperatures is required to obtain reasonable acceptance rates so that the extra temperature hardly matters.

An advantage of tempered transitions is that it is easy to code because one iteration basically consists of a loop of MCMC steps in which only the temperature and the step size need to be adjusted. For illustration, let us assume that we update x at temperature β by a normal proposal whose standard deviation σ varies with β. The tth iteration of tempered transitions can then be described by the following pseudo-code:

Algorithm 4.2:

Set $x = X_{t-1}$.
$denominator = 1$.
$numerator = 1$.
for(j in $1 : n$) // heating-up
 {
  $denominator = denominator \times p_{\beta_{j-1}}(x)$.
  $numerator = numerator \times p_{\beta_j}(x)$.
  Draw $x' \sim N\left(x, \sigma_j^2\right)$.

With probability $\alpha(x, x') = \min\left\{1, \frac{p_{\beta_j}(x')}{p_{\beta_j}(x)}\right\}$ set $x = x'$.

}

for(j in $n : 1$) // cooling-down

{

Draw $x' \sim N\left(x, \sigma_j^2\right)$.

With probability $\alpha(x, x') = \min\left\{1, \frac{p_{\beta_j}(x')}{p_{\beta_j}(x)}\right\}$ set $x = x'$.

$denominator = denominator \times p_{\beta_j}(x)$.

$numerator = numerator \times p_{\beta_{j-1}}(x)$.

}

With probability $\alpha = \min\left\{1, \frac{numerator}{denominator}\right\}$ set $X_t = x$, else set $X_t = X_{t-1}$.

It is not straightforward to see that the general tempered transitions algorithm (Algorithm 4.1) satisfies the detailed balance condition. Let us therefore verify the reversibility of the algorithm in the next section.

4.2.3 Reversibility

Neal proves that the tempered transitions algorithm is reversible for discrete state spaces. We will extend this proof to general state spaces. We will show that every particular path is reversible and then deduce that the transition from x_0 to x'_0 is also reversible, independent of the particular path.

We can verify that each path is reversible by checking that tempered transitions is a Metropolis-Hastings algorithm with equilibrium distribution $p_{\beta_0}(x)$. In tempered transitions, a particular path $(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)$ is drawn from the following proposal distribution

$$\begin{aligned}
&q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) \\
&\quad = T_{\beta_1}(x_0, x_1) \cdots T_{\beta_n}(x_{n-1}, x_n) \cdot T_{\beta_n}(x_n, x'_{n-1}) \cdot T_{\beta_{n-1}}(x'_{n-1}, x'_{n-2}) \cdots T_{\beta_1}(x'_1, x'_0) \\
&\quad = \left[\prod_{i=0}^{n-1} T_{\beta_{i+1}}(x_i, x_{i+1})\right] T_{\beta_n}(x_n, x'_{n-1}) \left[\prod_{i=0}^{n-2} T_{\beta_{i+1}}(x'_{i+1}, x'_i)\right].
\end{aligned}$$

Similarly, the reverse path $(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0)$ has the proposal distribution

$$\begin{aligned}
&q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0) \\
&\quad = T_{\beta_1}(x'_0, x'_1) \cdots T_{\beta_{n-1}}(x'_{n-2}, x'_{n-1}) \cdot T_{\beta_n}(x'_{n-1}, x_n) \cdot T_{\beta_n}(x_n, x_{n-1}) \cdots T_{\beta_1}(x_1, x_0) \\
&\quad = \left[\prod_{i=0}^{n-1} T_{\beta_{i+1}}(x_{i+1}, x_i)\right] T_{\beta_n}(x'_{n-1}, x_n) \left[\prod_{i=0}^{n-2} T_{\beta_{i+1}}(x'_i, x'_{i+1})\right].
\end{aligned}$$

If tempered transitions is truly a Metropolis-Hastings algorithm, then the Metropolis-Hastings acceptance probability

$$\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) = \min\left\{1, \frac{p_{\beta_0}(x'_0)\, q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0)}{p_{\beta_0}(x_0)\, q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)}\right\} \tag{4.3}$$

will be identical to the tempered transitions acceptance probability (4.1). For verifying that the acceptance probabilities are identical, we need that, for every $i = 1, \ldots, n$,

$$\frac{T_{\beta_i}(x, x')}{T_{\beta_i}(x', x)} = \frac{p_{\beta_i}(x')}{p_{\beta_i}(x)} \qquad \text{for all } x, x'. \tag{4.4}$$

This result comes from the detailed balance condition

$$p_{\beta_i}(x)\, T_{\beta_i}(x, x') = p_{\beta_i}(x')\, T_{\beta_i}(x', x) \qquad \text{for all } x, x'.$$

We can now express the acceptance probability (4.3) by

$$\begin{aligned}
&\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) \\
&\quad = \min\left\{1, \frac{p_{\beta_0}(x'_0)\, q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0)}{p_{\beta_0}(x_0)\, q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)}\right\} \\
&\quad = \min\left\{1, \frac{p_{\beta_0}(x'_0)}{p_{\beta_0}(x_0)} \left[\prod_{i=0}^{n-1} \frac{T_{\beta_{i+1}}(x_{i+1}, x_i)}{T_{\beta_{i+1}}(x_i, x_{i+1})}\right] \frac{T_{\beta_n}(x'_{n-1}, x_n)}{T_{\beta_n}(x_n, x'_{n-1})} \left[\prod_{i=0}^{n-2} \frac{T_{\beta_{i+1}}(x'_i, x'_{i+1})}{T_{\beta_{i+1}}(x'_{i+1}, x'_i)}\right]\right\} \\
&\quad \overset{(4.4)}{=} \min\left\{1, \frac{p_{\beta_0}(x'_0)}{p_{\beta_0}(x_0)} \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_{i+1}}(x_{i+1})}\right] \frac{p_{\beta_n}(x_n)}{p_{\beta_n}(x'_{n-1})} \left[\prod_{i=0}^{n-2} \frac{p_{\beta_{i+1}}(x'_{i+1})}{p_{\beta_{i+1}}(x'_i)}\right]\right\} \\
&\quad = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right] \left[\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\},
\end{aligned}$$

which is identical to the acceptance probability (4.1). Hence, tempered transitions is of Metropolis-Hastings form, and every path is reversible. The Metropolis-Hastings form also implies that, for a given proposal distribution q, tempered transitions is as efficient as possible (Peskun-optimality).

We can now deduce the desired reversibility of the transition x_0 to x'_0 (independent of a particular path) by integrating over all possible paths between x_0 and x'_0. The result follows from the reversibility of each particular path and from the interchangeability of integrals [Fubini's theorem (see for example Bauer 2001)]:

$$\begin{aligned}
&\int_A \mu(\mathrm{d}x_0) \int_\Omega \mu(\mathrm{d}x_1) \cdots \int_\Omega \mu(\mathrm{d}x_n) \int_\Omega \mu(\mathrm{d}x'_{n-1}) \cdots \int_\Omega \mu(\mathrm{d}x'_1) \int_B \mu(\mathrm{d}x'_0) \\
&\qquad p_{\beta_0}(x_0)\, q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) \min\left\{1, \frac{p_{\beta_0}(x'_0)\, q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0)}{p_{\beta_0}(x_0)\, q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)}\right\} \\
&= \int_B \mu(\mathrm{d}x'_0) \int_\Omega \mu(\mathrm{d}x'_1) \cdots \int_\Omega \mu(\mathrm{d}x'_{n-1}) \int_\Omega \mu(\mathrm{d}x_n) \cdots \int_\Omega \mu(\mathrm{d}x_1) \int_A \mu(\mathrm{d}x_0) \\
&\qquad p_{\beta_0}(x'_0)\, q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0) \min\left\{1, \frac{p_{\beta_0}(x_0)\, q(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)}{p_{\beta_0}(x'_0)\, q(x'_0, \ldots, x'_{n-1}, x_n, \ldots, x_0)}\right\}.
\end{aligned}$$

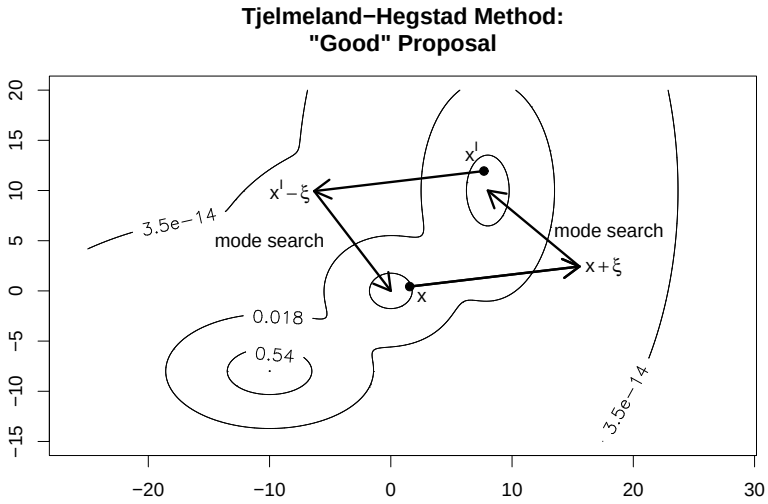

Figure 4-4: Mode jumping via local optimisation first takes a large step $x+\xi$ away from the current state x and then finds a new mode from which it draws a proposal state x'. For reversibility, the reverse step has to be carried out: jump away from x' to $x' - \xi$ and climb the nearest mode. Here, the sampler finds its way back to the mode of the current state x so that the acceptance probability of moving from x to x' will most likely be high. If it does not find back (as shown in Figure 4-5), the acceptance probability of moving from x to x' will most likely be very low.

4.3 Mode jumping via local optimisation

4.3.1 Algorithm

Mode jumping via local optimisation is based on deterministic mode searches (Tjelmeland and Hegstad 2001, Tjelmeland and Eidsvik 2004). If the sampler is currently in x, it will take first a large step to $x + \xi$ and then find the nearest mode by a deterministic mode search. If a new mode is found, it is approximated by a normal distribution from which a proposal state x' is generated. For reversibility of the algorithm, the reverse step has to be carried out under the assumption that the mode of x is unknown: first the sampler moves from x' to $x' - \xi$, then it searches for the closest mode and approximates this mode by a normal distribution. It then assumes that x is generated under the latter normal distribution. This assumption is important for the acceptance probability. If the method really finds its way back to the mode of the current state (see Figure 4-4), the acceptance probability of moving from x to x' will most likely be high. Otherwise, if it finds a third mode under which the current

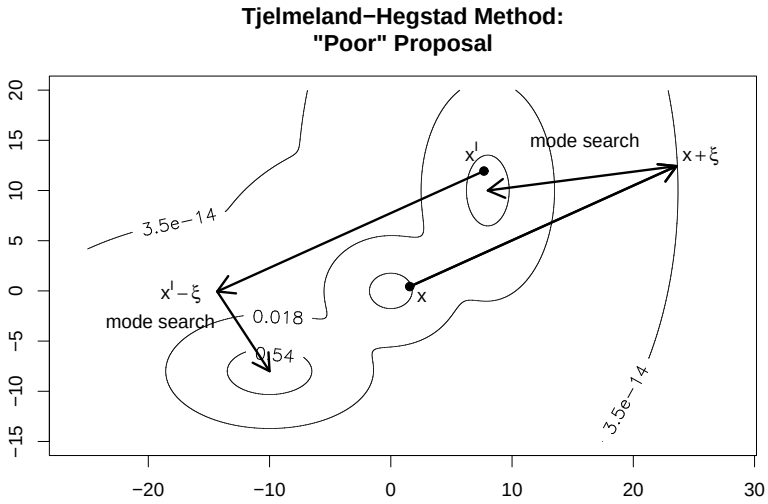

Figure 4-5: Mode jumping via local optimisation first takes a large step $x+\xi$ away from the current state x and then finds a new mode from which it draws a proposal state x'. For reversibility, the reverse step has to be carried out: jump away from x' to $x' - \xi$ and climb the nearest mode. Here the sampler does not find back to the mode of the current state x so that the acceptance probability will most likely be very low.

state x is unlikely (see Figure 4-5), the acceptance probability of moving from x to x' will most likely be very small.

We will here describe mode jumping via local optimisation as an auxiliary variable method (Sharp 2003). If X is the variable of interest with density $p(x)$ on $\Omega = \mathbb{R}^d$, then the step size ξ for the large jump is modelled as an independent auxiliary variable defined by some standard density $f(\xi)$, such as a normal distribution, on Ω. The joint distribution is then

$$\pi(x, \xi) = p(x)\, f(\xi)$$

on $\Omega \times \Omega$. Sampling from the joint distribution and discarding the ξ values produces a sample from $p(x)$ because the the joint distribution $\pi(x, \xi)$ has marginal distribution $p(x)$.

For reversibility, the large step should be invertible. Here we will define the large step by $t: \; x \mapsto (x + \xi)$ and its inverse by $t^{-1}: \; x \mapsto (x - \xi)$, but other definitions of invertible transformations t are possible. For reversibility, it is also important that the direction of the large step, either $(x + \xi)$ or $(x - \xi)$, is drawn with equal probability. If the large step away from the current state x

is chosen to be $x \mapsto (x + \xi)$, then the reverse step away from the proposal x' has to be $x' \mapsto (x' - \xi)$; otherwise if the large step away from x is $x \mapsto (x - \xi)$, then the reverse step away from the proposal x' has to be $x' \mapsto (x' + \xi)$. Based on these specifications, each iteration of mode jumping via local optimisation consists of the following steps:

Algorithm 4.3:

Step 1 Set $x = X_{t-1}$.

Step 2 Draw $\xi \sim f(\xi)$ independently of x.

Step 3 Conditional on ξ, define the proposal distributions

$$q_0(x, x') \sim N\Big(\mu(x + \xi), \Sigma(x + \xi)\Big) \qquad \text{and}$$
$$q_1(x, x') \sim N\Big(\mu(x - \xi), \Sigma(x - \xi)\Big)$$

where $\mu(z)$ denotes the mode of $p(x)$ found by a local optimisation algorithm started at z and where $\Sigma(z)$ denotes the inverse of the Hessian matrix of $p(x)$ at $\mu(z)$. Recall that the Hessian matrix H is a $d \times d$ matrix containing the second partial derivatives. Its entries are $H_{ij} = \frac{\partial^2 p(x)}{\partial x_j \partial x_i}$.

Step 4 Draw $i = 0, 1$ with probability $\frac{1}{2}$. Then draw $x' \sim q_i(x, x')$ and calculate

$$\alpha_{i,1-i}(x, x') = \min\left\{1, \frac{p(x')q_{1-i}(x', x)}{p(x)q_i(x, x')}\right\}. \tag{4.5}$$

Step 5 Draw $u \sim U(0, 1)$. If $u < \alpha_{i,1-i}(x, x')$, then accept the move and set $X_t = x'$; else, remain in the current state and set $X_t = x$.

Recall that $q_{1-i}(x', x)$ is the normal distribution which approximates the mode found by the reverse step. If the reverse step finds a mode under which the current state x is unlikely, then the value $q_{1-i}(x', x)$ will be very small and thus the acceptance probability (4.5) will most likely also be very small. In the toy example below (Section 4.4), actually all the rejections are due to missing the original mode on the reverse step. As this proportion is quite large (76.6%), this is a significant drawback of the method.

4.3.2 Optimality of acceptance probability

Tjelmeland and Hegstad (2001) state that the design of the x update follows the general form of the Metropolis-Hastings algorithm in which the acceptance

probability can be expressed by $\alpha(x,x') = s(x,x')/\left[1+\frac{p(x')\,q(x',x)}{p(x)\,q(x,x')}\right]$ where $s(x,x')$ is symmetric and such that $0 \leq \alpha(x,x') \leq 1$. In their algorithm, the proposal distribution is the mixture of distributions

$$q(x,x') = \frac{1}{2}\,q_0(x,x') + \frac{1}{2}\,q_1(x,x')$$

and the symmetric function is

$$\begin{aligned} s(x,x') =& \frac{1}{2}\,\left[p(x)\,q_0(x,x')\,\alpha_{0,1}(x,x') + p(x')\,q_0(x',x)\,\alpha_{0,1}(x',x)\right] \\ & \times \left[\frac{1}{p(x)\,q(x,x')} + \frac{1}{p(x')\,q(x',x)}\right]. \end{aligned}$$

The general Metropolis-Hastings form proves the validity of their MCMC algorithm. The disadvantage of this general form is that the acceptance probability is not Peskun-optimal. That means that it does not minimise the integrated autocorrelation time for the given proposal distribution q. Let us compare this sub-optimal acceptance probability in the original form (4.5)

$$\alpha_{i,1-i}(x,x') = \min\left\{1, \frac{p(x')q_{1-i}(x',x)}{p(x)q_i(x,x')}\right\}$$

with the (hypothetical) optimal acceptance probability

$$\begin{aligned} \alpha(x,x') &= \min\left\{1, \frac{p(x')\,q(x',x)}{p(x)\,q(x,x')}\right\} \\ &= \min\left\{1, \frac{p(x')\,\left[\frac{1}{2}\,q_0(x',x) + \frac{1}{2}\,q_1(x',x)\right]}{p(x)\,\left[\frac{1}{2}\,q_0(x,x') + \frac{1}{2}\,q_1(x,x')\right]}\right\}. \end{aligned}$$

When justifying the sub-optimal choice, Tjelmeland and Hegstad point out that their acceptance probability requires calculating only two proposal kernels, namely $q_{1-i}(x',x)$ and $q_i(x,x')$ (for i fixed), while the optimal acceptance probability needs all four proposal kernels $q_0(x',x)$, $q_1(x',x)$, $q_0(x,x')$ and $q_1(x,x')$. As evaluating a single proposal kernel is very expensive due to the local optimisation involved, Tjelmeland and Hegstad prefer their proposal type for computational efficiency. There is another point in their favour which has not been seen so far. If all the modes of the target distribution are isolated, then it is very likely that the nearest mode to $(x+\xi)$ is different (and thus separated) from the nearest mode to $(x-\xi)$ in which case the proposal distribution

$$q(x,x') = \frac{1}{2}\,q_0(x,x') + \frac{1}{2}\,q_1(x,x')$$

has two isolated modes, one under q_0, the other under q_1. If move type i is chosen, then $q(x,x') \approx \frac{1}{2}\,q_i(x,x')$ and $q(x',x) \approx \frac{1}{2}\,q_{1-i}(x',x)$ so that there is probably not much difference between the optimal acceptance probability $\alpha(x,x') = \min\left\{1, \frac{p(x')\,q(x',x)}{p(x)\,q(x,x')}\right\}$ and the sub-optimal acceptance probability $\alpha_{i,1-i}(x,x') = \min\left\{1, \frac{p(x')q_{1-i}(x',x)}{p(x)q_i(x,x')}\right\}$, in which case Tjelmeland and Hegstad's mode jumping algorithm is close to Peskun-optimal.

4.3.3 Avoiding numerical instability

One issue with mode jumping via local optimisation is that the local optimisation algorithm may be numerically instable when started in a low-probability area. This instability arises from small density values $p(x)$ being computationally equal to zero. We may still be able to distinguish two otherwise computationally indistinguishable small values $p(x)$ and $p(y)$ by working with their natural logarithms $\log[p(x)]$ and $\log[p(y)]$. This is probably the reason why Tjelmeland and Hegstad suggest finding the modes of the target density by finding the maximisers of the log-function $\log[p(x)]$, which is an equivalent problem. Unfortunately, working with log-probabilities does not entirely remove the problem of numerical instability because the calculation of log-probabilities, if carried out in a naive way, is also prone to numerical instability. We will discuss this problem in the following.

Suppose the target distribution $p(x)$ is a mixture of distributions of the general form

$$p(x) \propto \sum_{k=1}^{m} c_k \exp[-h_k(x)] = \sum_{k=1}^{m} \exp[-(h_k(x) - \log c_k)]$$

where $\{c_k\}$ are constants and $\{h_k(x)\}$ are the so-called energy functions. Then the log-function is defined by

$$\log[p(x)] = \log\left\{\sum_{k=1}^{m} \exp[-(h_k(x) - \log c_k)]\right\}.$$

If the state x lies in a low-probability area, then every $\exp[-(h_k(x) - \log c_k)]$, $k = 1, \ldots, m$, will most likely be set computationally equal to zero so that their sum will also be zero. In consequence, the optimisation algorithm cannot distinguish between two very small probabilities and fails. To avoid that the sum is set computationally equal to zero, we can define

$$\gamma(x) = -\min_{1 \le k \le m} \{h_k(x) - \log c_k\}$$

and use the representation

$$\begin{aligned}\log[p(x)] =& \log\left\{\exp[\gamma(x)] \sum_{k=1}^{m} \exp[-(h_k(x) - \log c_k) - \gamma(x)]\right\} \\ =& \gamma(x) + \log\left\{\sum_{k=1}^{m} \exp[-(h_k(x) - \log c_k) - \gamma(x)]\right\}\end{aligned}$$

(Tjelmeland 2005, personal communication). By definition of $\gamma(x)$, there is at least one k such that $\exp[-(h_k(x) - \log c_k) - \gamma(x)] = 1$ so that

$$\log\left\{\sum_{k=1}^{m} \exp[-(h_k(x) - \log c_k) - \gamma(x)]\right\} \ge 0$$

and thus

$$\log[p(x)] \geq \gamma(x)\,.$$

As $\gamma(x)$ depends on the x value, we can computationally distinguish the log-probabilities at two different states x and y even if both states lie in a low-probability area so that we can find the maxima of $\log[p(x)]$ from all starting points. Similar ideas can be applied to gain other numerically stable functions, for example the first and second derivatives of $\log[p(x)]$, if needed.

4.4 Comparison on a toy example

4.4.1 Toy example

For comparison, mode jumping via local optimisation and tempered transitions will be tested on the toy problem suggested by Tjelmeland and Hegstad, namely on sampling from a mixture of thirteen bivariate normal distributions of equal weight and shape whose modes are isolated. To complicate the mixing between modes, the mass of each mode is concentrated on a small spot and thus difficult to hit by standard Metropolis-Hastings algorithms. The thirteen modes are arranged symmetrically on an "outer" and an "inner" square both centred at the origin. Moreover, the distance between the origin and the "inner" square is a tenth of the distance between the origin and the "outer" square. This difference in scaling adds to the complexity of the problem. To visualise the arrangement of modes, it may help to look at Figure 4-6 (although the main reason for this figure is to show the results of the experiment). In mathematical terms, the bivariate sampling problem is given by

$$p(x) \sim \sum_{k=1}^{13} \frac{1}{13}\, N(\mu_k, \Sigma_k)\,, \qquad x \in \mathbb{R}^2,$$

where the mode $\mu_1 = (0, 0)$ occupies the origin, the modes $\mu_2, \ldots, \mu_5$ lie on the edges of the "inner" square, namely on $(0.1, 0.1)$, $(-0.1, 0.1)$, $(0.1, -0.1)$, $(-0.1, -0.1)$, respectively, and the modes $\mu_6, \ldots, \mu_{13}$ are located on both the edges and the midpoints of the "outer" square boundary, namely on $(1, 1)$, $(1, 0)$, $(1, -1)$, $(0, 1)$, $(0, -1)$, $(-1, 1)$, $(-1, 0)$, $(-1, -1)$, respectively. The diagonal elements of the covariance matrices Σ_k, $k = 1, \ldots, 13$, are equal to 0.01^2, while the off-diagonal elements are equal to zero.

In the next sections, we will specify both mode jumping algorithms for this particular toy example before presenting the results of the experiment.

4.4.2 Implementing mode jumping via local optimisation

The set-up of mode jumping via local optimisation is here very similar to the original set-up of the sampler (Tjelmeland and Hegstad 2001). Following the original specification, the auxiliary distribution $f(\xi)$ which determines the size of the large step at each iteration is bivariate normal with zero mean and diagonal covariance matrices with diagonal elements equal to $\sigma_\xi^2 = 2^2$, while the deterministic mode search is based on a Newton method [here the Newton-Raphson method as described in Walsh (1975, Section 4.3)]. Furthermore, the initial state is drawn from the target distribution, so that no burn-in is needed. The Markov chain is run for $N = 100\,000$ iterations. Here, each iteration consists of a single mode jumping step via local optimisation. Originally, one iteration was based on one mode jumping step and 250 simple Metropolis-Hastings steps designed for the local exploration of the modes. These local steps are left out here because they do not contribute to the mixing between modes, which is the subject of this study.

4.4.3 Implementing tempered transitions

The set-up of the tempered transitions algorithm for the toy problem was specified by trial and error. The hottest distribution $p_{\beta_n}(x) \propto [p(x)]^{\beta_n}$ has minimal inverse temperature $\beta_n = \frac{1}{400}$. For comparability with the mode jumping via local optimisation algorithm in which the large step away from the current mode has variance $\sigma_\xi^2 = 2^2$, the variance of the proposal step at the hottest temperature in tempered transitions is set to $\sigma_n^2 = 2^2$ so that, at the hottest temperature, the jth component of the proposal state is generated by $x'_j \sim N(x, \sigma_n^2)$ for $j = 1, 2$. Updating x component-wise gives much better mixing than updating the components of x jointly under the hottest distribution. (The acceptance rates are 32.3% and 12.8% respectively.) As the mixing under the hottest distribution is crucial, the component-wise update is chosen. We also have to specify a scheme for the inverse temperatures $\{\beta_i\}$ and a plan for the proposal step sizes $\{\sigma_i\}$. Since proposals for the component x_j, $j = 1, 2$, are drawn from a normal distribution $x'_j \sim N(x_j, \sigma_i^2)$, $i = 1, \ldots, n$, it seems reasonable to let the step size σ_i grow in the same way in which the standard deviation of a normal distribution would expand when the distribution is heated up. Before we can imitate this behaviour, we have to describe it. Consider the two tempered normal distributions $N(0, \rho^2/\beta_{i+1})$ and $N(0, \rho^2/\beta_i)$. The difference

$\left(\sqrt{\rho^2/\beta_{i+1}} - \sqrt{\rho^2/\beta_i}\right)$ between their standard deviations is proportional to $\left(\sqrt{1/\beta_{i+1}} - \sqrt{1/\beta_i}\right)$. For mirroring this behaviour, we will require that the tempered transitions step size plan $\{\sigma_i\}$ satisfies $(\sigma_{i+1} - \sigma_i) \propto \left(\sqrt{1/\beta_{i+1}} - \sqrt{1/\beta_i}\right)$. Furthermore, we want the tempered transition step size plan $\{\sigma_i\}_{i=1}^n$ to range from the coldest step size $\sigma_1 = 0.01$, which allows local exploration, to the hottest step size $\sigma_n = 2$, which allows global exploration. We can account for this by setting

$$\sigma_i := \sigma_1 + \frac{\sigma_n - \sigma_1}{\sqrt{1/\beta_n} - \sqrt{1/\beta_1}} \left(\sqrt{1/\beta_i} - \sqrt{1/\beta_1}\right), \qquad i = 1, 2, \ldots, n.$$

This definition also satisfies $(\sigma_{i+1} - \sigma_i) \propto \left(\sqrt{1/\beta_{i+1}} - \sqrt{1/\beta_i}\right)$ as required. It remains to choose the inverse temperatures. Here, we will follow the standard advice of spacing temperatures geometrically by $\beta_i = \beta_{\min}^{(i-1)/(n-1)}$, $i = 1, 2, \ldots, n$, and $\beta_0 = 1$ by definition. Due to the hard sampling problem, many temperatures ($n = 400$) are needed to obtain a reasonable acceptance rate for the tempered transitions path, but the algorithm gains speed if only one of the components of x is updated at random (with probability $\frac{1}{2}$) at each temperature.

4.4.4 Results

Before we can compare the algorithms, we have to find the criteria for the comparison. We are interested in the mixing between modes. It is therefore helpful to monitor which of the 13 modes $\mu_1, \ldots, \mu_{13}$ is currently visited. Suppose x is the current state, then the mode index function

$$z(x) = \text{argmin}_{k=1,2,\ldots,13} \{\|x - \mu_k\|\}$$

determines the nearest mode to x based on the Euclidean norm. For instance, if $z(x) = 9$, then the sample x comes from the mode $\mu_9 = (0, 1)$. As all the modes should be visited equally often, the theoretical expectation of the mode index is

$$\begin{aligned} \mathbb{E}_{p(x)}\left[z(X)\right] &= \frac{1}{13} \sum_{k=1}^{13} k \\ &= 7\,. \end{aligned}$$

One way of measuring the mixing between modes is estimating the integrated autocorrelation time $\tau(Z) = \sum_{t=-\infty}^{\infty} \rho_t(Z)$ of the process $\{z(X_i)\}$ by Geyer's positive sequence estimator (2.3). We can improve the accuracy of this estimator by calculating the sample autocorrelations $\{\rho_t(Z)\}$ with respect to

the theoretical mean $\mathbb{E}_{p(x)}[z(X)]$ (rather than the empirical one). Another way of assessing the mixing between modes is determining the average length of a single visit to a particular mode. The length of a mode visit is the number of iterations which the sampler spends in the mode before leaving it again; if the sampler enters a particular mode in the first iteration and leaves it after the sixth iteration, then the length of the mode visit is 6. The average length can be inferred from the mode index process $\{z(X_i)\}$. Suppose the mode index process is given by $\{z(X_i)\} = \{1, 1, 1, 2, 1, 2, 2, 3, 2, 3\}$. We can see that the sampler visits the first mode twice, first for three iterations and then for one iteration which gives an average length of two iterations per visit. Similarly, we can deduce that the sampler spends on average $\frac{4}{3}$ iterations per visit in the second mode and one iteration per visit in the third mode. Other important criteria are the acceptance rate and the computational cost of the algorithms.

First let us compare the integrated autocorrelation time $\tau(Z)$ which measures the mixing quality of the algorithm. Mode jumping via local optimisation gives an estimated integrated autocorrelation time of $\hat{\tau}_{\mathrm{TH}} = 52.74$ which is about 15 times greater than the autocorrelation time $\hat{\tau}_{\mathrm{TT}} = 3.63$ in tempered transitions. This means that tempered transitions mixes much better between modes than the Tjelmeland-Hegstad algorithm. Figures 4-6 and 4-7 display the average visit lengths to each mode. These figures confirm that tempered transitions mixes better between modes. It spends on average an equal amount of time (2.6 iterations) in each mode before leaving it again, independently on the location of the mode, while the Tjelmeland-Hegstad method has difficulties in escaping modes that are closely surrounded by other modes. Mode jumping via local optimisation needs on average 114 iterations to leave the central mode μ_0, approximately 10 iterations to escape from the modes at the edges of the inner square $(\mu_1, \ldots, \mu_5)$, 3.7 iterations to jump away from the midpoints of the outer square $(\mu_7, \mu_9, \mu_{10}, \mu_{12})$ and 3.1 iterations to exit the edges of the outer square $(\mu_6, \mu_8, \mu_{11}, \mu_{13})$. As the sampler does not leave a mode if the mode jumping proposal is rejected, monitoring the reason for the rejections helps in understanding the asymmetry in the mode visit lengths. Whenever a mode jump was rejected, it was because the reverse jump in the mode jumping proposal did not find its way back to the original mode. This problem was illustrated earlier in Figure 4-5. It also explains why the average visit lengths vary with the location of the modes. The mode at the origin is closely and completely surrounded by the inner square modes so that it is quite hard to

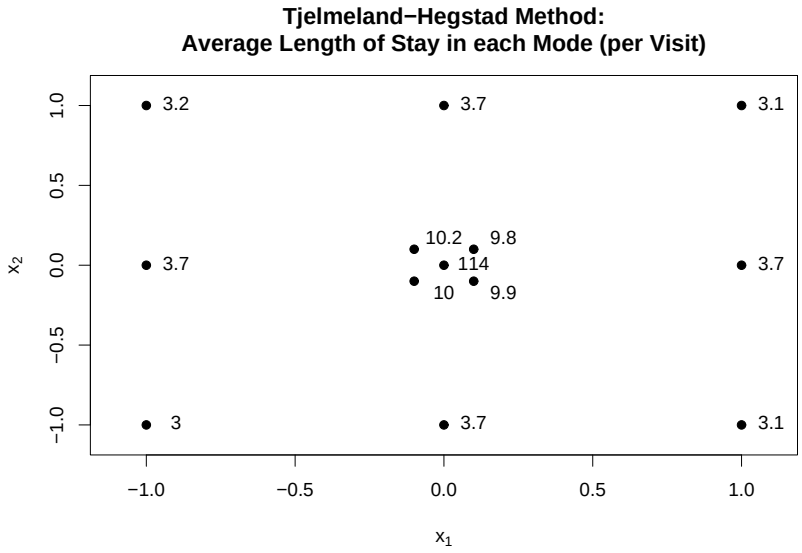

Figure 4-6: The Tjelmeland-Hegstad method (mode jumping via local optimisation) has difficulties escaping from modes which are closely surrounded by other modes. The average times the sampler spends in each mode per visit are shown above.

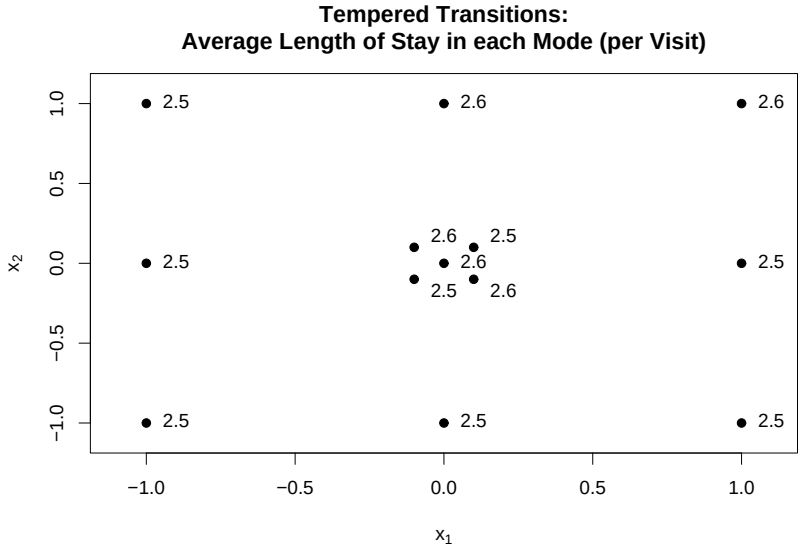

Figure 4-7: Tempered transitions does not have any problems escaping modes. It spends on average an equal amount of time in each mode per visit.

find this mode on the jump back from any other mode. Similarly, the inner square modes have themselves and the origin as close neighbours. They are however not completely surrounded so that their basin of attraction is greater than that of the origin mode, which explains the smaller average visit length. The even greater distance between the outer square modes leads again to a smaller average length of visit in the outer modes. It is interesting that the position of the outer square modes matters. It is easier to leave the edges than the midpoints. The reason is probably that the edge modes have two direct neighbours (the nearest midpoint modes), while the midpoint modes have three direct neighbours (the two nearest edge modes and the mode group at the centre). If we included the diagonal neighbours as direct neighbours into the counting, we would also come to the conclusion that the midpoint modes have more neighbours than the edge modes and are therefore more difficult to find on the reverse jump. It remains to compare the acceptance rate and the computing time of the algorithms. Again, tempered transitions is the superior method because it has a higher acceptance rate (42.8%) than mode jumping via local optimisation (23.4%) and because it takes only 56% of the computing time of the other algorithm (140 minutes versus 250 minutes).

In summary, we have seen an example where tempered transitions is almost twice as fast as the Tjelmeland-Hegstad method and mixes far better between modes. While tempered transitions leaves modes quite quickly independent of the mode location, mode jumping via local optimisation is not that quick. It will be stuck in a mode longer, the more neighbouring modes there are and the closer they are. Mode jumping via local optimisation also has the disadvantage of being prone to numerical instability so that cumbersome precautions need to be taken. Tempered transitions, on the other hand, is easy to implement. The only difficulty is the tuning of temperatures. The choice of temperatures is important for the efficiency of the algorithm. So far, temperatures have been tuned by trial and error and without claim of optimality. It would be helpful if we could optimise the temperatures by a systematic tuning approach. We will develop such an approach in the next chapter.

Chapter 5

Tuning Temperatures in Tempered Transitions

5.1 Introduction

In Section 5.2, we will discuss the aim of tuning temperatures in tempered transitions. If we assume that the hottest temperature and the number of temperatures are fixed, the aim of optimisation is maximising the expected acceptance probability. As we will see, we are unable to calculate (or estimate) the expected acceptance probability in most cases so that we cannot maximise it. On the search for an alternative approach, we will find that the expected acceptance probability is connected to another expected value which we can calculate (or estimate) under idealising assumptions. Optimising the temperatures with respect to the other idealised expectation will help us in improving the acceptance probability even if the idealising assumptions are not met. In Section 5.3, we will derive some theoretical properties of the optimisation problem which will help us solve it. We can then discuss possible analytic and numerical optimisation approaches in Section 5.4. The results will be summarised in Section 5.5.

5.2 Posing the optimisation problem

5.2.1 How to improve the efficiency of tempered transitions

So far, we have only stated that we can improve the efficiency of tempered transitions by tuning temperatures. Now we will justify this claim. A cost-

efficient algorithm is an algorithm that mixes well at a low cost. Let us start with the mixing. The mixing of tempered transitions depends on the mode jumping ability of the algorithm. This ability depends on the flexibility of the proposal mechanism and, if a mode jump is proposed, on the acceptance probability. As discussed in Section 4.2.2, the proposal mechanism of tempered transitions starts a secondary chain which passes through the temperatures first in increasing order (heating-up) and then in decreasing order (cooling-down) to generate a proposal for tempered transitions. For full flexibility of the proposal mechanism, the hottest temperature should be chosen such that the secondary chain can move freely around the sample space at this temperature. The fast mixing at the hottest temperature allows the secondary chain to move to a different part of the sample space and to find another mode of the target distribution in the cooling-down process. For good mixing, it is not enough that the sampler is able to generate mode jumping proposals. If the sampler keeps rejecting the proposals, it will hardly mix between modes. For good mixing, we also need "reasonable" acceptance probabilities. But how do we define "reasonable"? In standard MCMC, it is often recommended to aim for an average acceptance probability (acceptance rate) of 20% to 40%. This recommendation is based on the assumption that we draw the proposal state from a unimodal distribution centred at the current state. For ease of presentation, let us choose a normal proposal $x' \sim N(x, \sigma^2)$. The acceptance probability is then $\alpha(x, x') = \min\left\{1, \frac{\pi(x')}{\pi(x)}\right\}$. If the step size σ is too small, the states x and x' and thus the density values $\pi(x)$ and $\pi(x')$ lie close together so that the acceptance probability is high. Although the algorithm moves, it cannot move far so that it mixes very slowly within one mode and hardly between modes. If the step size is too big, the step from x to x' will quite likely land in a low-probability area. This means that the acceptance probability is very small so that the algorithm rarely moves and therefore mixes poorly. As a result, very small and very high average acceptance probabilities indicate poor mixing so that the "reasonable" acceptance probabilities are the ones in-between. This definition may however change with the example. Consider the case of drawing an independent proposal from an approximation $q(x)$ to the target distribution $\pi(x)$ and accepting the proposal with probability $\alpha(x, x') = \min\left\{1, \frac{\pi(x')\,q(x)}{\pi(x)\,q(x')}\right\}$. If the approximation is poor, then the proposals will often land in low-probability areas and thus often produce small acceptance probabilities so that small acceptance probabilities are again a sign of poor mixing. If, on the other hand, the approximation is quite good, then $q(x)$ will

approximately cancel with $\pi(x)$ in the acceptance ratio and, similarly, $q(x')$ will approximately cancel with $\pi(x')$ so that high acceptance probabilities are achieved. In this case, high acceptance probabilities are a sign of fast mixing and therefore desirable. We learn from these examples that a "reasonable" average acceptance probability cannot be defined in absolute terms. We could say that an average acceptance probability is "reasonable" if it indicates good mixing of the underlying sampler. But which acceptance probabilities indicate good mixing in tempered transitions? The answer depends on the mixing ability of the secondary chain. Suppose the mixing ability is poor perhaps because the number of temperatures is too low so that the secondary chain does not take enough steps, or perhaps because the hottest temperature is not hot enough so that the chain does not leave the current mode, then the mixing of tempered transitions will be poor no matter how high the acceptance probabilities are. If, on the other hand, the mixing ability of the secondary chain is good, then the mixing of tempered transitions will be better, the higher the average acceptance probability. The interim conclusion is that we can improve the mixing of tempered transitions by maximising its average acceptance probability, provided that the secondary chain in tempered transitions mixes fast around the sample space. As the acceptance probability

$$\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right] \left[\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\}$$

depends on the temperatures $\{\beta_i\}_{i=0}^n$, the choice of temperatures will affect the average acceptance probability although, at this point, it is not clear to what degree. When defining the temperatures, we can decide on their number and their value. In particular, we have to choose the hottest temperature whose value is crucial for the mixing of the secondary chain. If it is not hot enough, the secondary chain will not be able to jump between modes. Another issue is that the length of the secondary chain generating a mode jumping proposal is proportional to the number of temperatures. If we increase the number of temperatures, we also increase the computational cost of tempered transitions. In practice, it can be observed that increasing the number of temperatures leads to higher average acceptance probabilities and thus to better mixing. As this improvement comes at a higher cost, we should only increase the number of temperatures if the gain outweighs the cost. The trade-off between mixing and cost and the constraint on the hottest temperature make it necessary to adjust the interim conclusion. The revised optimisation problem comes now in two parts: first, maximise the average acceptance probability under the constraint

that the hottest inverse temperature $\beta_{\min}$ and the number n of temperatures are fixed and provided that the secondary chain mixes well around the sample space; second, vary $\beta_{\min}$ (subject to being hot enough) and n only if the benefit is greater than the cost. In the next section, we will check the feasibility of such an approach.

5.2.2 Feasibility of the true optimisation problem

The average acceptance probability is an empirical value better known as the acceptance rate. One maximisation approach is therefore to run tempered transitions with various temperature choices and then to pick the choice that gave the highest acceptance rate. That is however exactly the approach we want to avoid because it is quite time-consuming. We would be quicker if we could calculate (or approximate) the theoretical value of the average acceptance probability, which is the expected acceptance probability. To derive the expected acceptance probability $\mathbb{E}_\varphi\left[\alpha(X_0,\ldots,X_{n-1},X_n,X'_{n-1},\ldots,X'_0)\right]$, we need the underlying distribution φ which is the joint distribution of the auxiliary path $(X_0,\ldots,X_{n-1},X_n,X'_{n-1},\ldots,X'_0)$ generated by the secondary chain. If we assume that tempered transitions has reached convergence, then the initial state X_0 is a sample from the target distribution p_{β_0}, while the other states follow the distribution of the Markov transition kernels $\{T_{\beta_i}\}_{i=1}^n$ under which they are generated so that the joint distribution $\varphi(X_0,\ldots,X_{n-1},X_n,X'_{n-1},\ldots,X'_0)$ is given by

$$\begin{aligned}&\varphi(x_0,\ldots,x_n,x'_{n-1},\ldots,x'_0)\\&\qquad= p_{\beta_0}(x_0)\left[\prod_{i=0}^{n-1}T_{\beta_{i+1}}(x_i,x_{i+1})\right]T_{\beta_n}(x_n,x'_{n-1})\left[\prod_{i=0}^{n-2}T_{\beta_{i+1}}(x'_{i+1},x'_i)\right].\end{aligned}\tag{5.1}$$

This yields the expected acceptance probability

$$\begin{aligned}&\mathbb{E}_\varphi\left[\alpha(X_0,\ldots,X_{n-1},X_n,X'_{n-1},\ldots,X'_0)\right]\\&\quad=\mathbb{E}_\varphi\left\{\min\left[1,\left(\prod_{i=0}^{n-1}\frac{p_{\beta_{i+1}}(X_i)}{p_{\beta_i}(X_i)}\right)\left(\prod_{i=0}^{n-1}\frac{p_{\beta_i}(X'_i)}{p_{\beta_{i+1}}(X'_i)}\right)\right]\right\}\\&\quad=\int_{\Omega^{2n+1}}\mathrm{d}\varphi(x_0,\ldots,x_n,x'_{n-1},\ldots,x'_0)\\&\qquad\qquad\min\left[1,\left(\prod_{i=0}^{n-1}\frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right)\left(\prod_{i=0}^{n-1}\frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right)\right]\\&\quad=\int_{\Omega^{2n+1}}p_{\beta_0}(\mathrm{d}x_0)\left[\prod_{i=0}^{n-1}T_{\beta_{i+1}}(x_i,\mathrm{d}x_{i+1})\right]T_{\beta_n}(x_n,\mathrm{d}x'_{n-1})\left[\prod_{i=0}^{n-2}T_{\beta_{i+1}}(x'_{i+1},\mathrm{d}x'_i)\right]\\&\qquad\qquad\min\left[1,\left(\prod_{i=0}^{n-1}\frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right)\left(\prod_{i=0}^{n-1}\frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right)\right].\end{aligned}$$

The integral form reveals that the expected acceptance probability depends on the choice of temperatures $\{\beta_i\}_{i=1}^n$ and on the choice of transition kernels $\{T_{\beta_i}\}_{i=1}^n$. We can simplify this optimisation problem by assuming that the Markov transition kernels $\{T_{\beta_i}\}_{i=1}^n$ produce independent samples from the respective equilibrium distributions $\{p_{\beta_i}\}_{i=1}^n$ and that the tempered transitions algorithm has reached convergence so that $X_0 \sim p_{\beta_0}$. In this case, the joint distribution is defined by

$$\begin{aligned} &\varphi(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) \\ &\qquad \propto p_{\beta_0}(x_0) \left[\prod_{i=1}^{n} p_{\beta_i}(x_i)\right] \left[\prod_{i=0}^{n-1} p_{\beta_{i+1}}(x'_i)\right]. \end{aligned} \tag{5.2}$$

We will refer to this assumption as the "ideal world" scenario to distinguish it from the "real world" scenario in which the general form of the joint distribution (5.1) cannot be reduced to (5.2). If we assume the "ideal world" scenario, then the expected acceptance probability simplifies to

$$\begin{aligned} &\mathbb{E}_\varphi \left[\alpha(X_0, \ldots, X_{n-1}, X_n, X'_{n-1}, \ldots, X'_0)\right] \\ &\quad = \int_{\Omega^{2n+1}} \prod_{i=0}^{n} p_{\beta_i}(\mathrm{d}x_i) \prod_{i=0}^{n-1} p_{\beta_{i+1}}(\mathrm{d}x'_i) \\ &\qquad \min\left[1, \left(\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right)\left(\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right)\right], \end{aligned}$$

which has the advantage that it depends solely on the choice of temperatures $\{\beta_i\}_{i=0}^n$ (and not anymore on the transition kernels $\{T_{\beta_i}\}_{i=1}^n$). In general, the expectation $\mathbb{E}_\varphi(\alpha)$ is neither in the "real world" nor in the "ideal world" tractable so that we cannot tackle the optimisation problem directly. However, as we will see in the next section, we may be able to tackle the problem implicitly by solving a related optimisation problem.

5.2.3 Searching for an alternative optimisation problem

We will develop an alternative optimisation problem based on an argument that Neal (1996) uses to explain why increasing the number of temperatures between the hottest and the coldest temperature improves the acceptance rate in tempered transitions when the tempered distributions follow the canonical form $p_\beta(x) \propto \exp[-\beta\, h(x)]$. We will adapt the argument slightly so that it can be applied to the wider class of tempered distributions

$$p_\beta(x) \propto \pi(x) \exp[-\beta\, h(x)] \tag{5.3}$$

because, as discussed in Section 4.2, this class provides a greater flexibility for the implementation of tempered transitions.

If the tempered distributions belong to the wider class (5.3), we can rewrite the tempered transitions acceptance probability by

$$\begin{aligned}
&\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) \\
&\quad = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right] \left[\prod_{i=0}^{n-1} \frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\} \\
&\quad = \min\left\{1, \left[\prod_{i=0}^{n-1} \frac{\pi(x_i)\exp\left[-\beta_{i+1}\, h(x_i)\right]}{\pi(x_i)\exp\left[-\beta_i\, h(x_i)\right]}\right] \left[\prod_{i=0}^{n-1} \frac{\pi(x'_i)\exp\left[-\beta_i\, h(x'_i)\right]}{\pi(x'_i)\exp\left[-\beta_{i+1}\, h(x'_i)\right]}\right]\right\} \\
&\quad = \min\left\{1, \exp\left[\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x_i)\right] \exp\left[-\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x'_i)\right]\right\} \\
&\quad = \min\left\{1, \exp\left[-\left(\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x'_i) - \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x_i)\right)\right]\right\}.
\end{aligned}$$

In this case, the acceptance probability depends on the size of the area under two step-functions. One of these areas

$$F_{\text{up}} := \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x_i)$$

is based on the states $\{x_i\}$ of the secondary chain generated in the heating-up process, while the other

$$F_{\text{down}} := \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, h(x'_i)$$

is based on the states $\{x'_i\}$ of the secondary chain generated in the cooling-down process. In the new notation, the acceptance probability becomes

$$\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) = \min\left\{1, \exp\left[-\left(F_{\text{down}} - F_{\text{up}}\right)\right]\right\}.$$

The acceptance probability will be greater, the smaller the difference between the areas F_{down} and F_{up}. To learn about this difference, let us look at the expected values

$$\begin{aligned}
\mathbb{E}_\varphi\left(F_{\text{up}}\right) &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, \mathbb{E}_\varphi\left[h(X_i)\right] \\
\text{and} \quad \mathbb{E}_\varphi\left(F_{\text{down}}\right) &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\, \mathbb{E}_\varphi\left[h(X'_i)\right]
\end{aligned}$$

with respect to the joint distribution $\varphi(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)$. If we assume the "ideal world" scenario (5.2) in which the Markov transition kernels $\{T_{\beta_i}\}_{i=1}^n$ generate independent samples from the corresponding distributions $\{p_{\beta_i}\}_{i=1}^n$ and tempered transitions has reached convergence, then the expectations

reduce to

$$\mathbb{E}_\varphi\left(F_{\text{up}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, \mathbb{E}_{p_{\beta_i}}\left[h(X)\right]$$
$$\text{and} \quad \mathbb{E}_\varphi\left(F_{\text{down}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, \mathbb{E}_{p_{\beta_{i+1}}}\left[h(X)\right].$$

This means that the expectations depend on the inverse temperatures $\{\beta_i\}$ entirely. To visualise the relationship between temperatures and expectations, we can use the "trick" of regarding the expectation of the energy function h at inverse temperature β as a function g of β and define

$$g(\beta) := \mathbb{E}_{p_\beta}\left[h(X)\right].$$

The "trick" reveals that the expectations

$$\mathbb{E}_\varphi\left(F_{\text{up}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, g(\beta_i)$$
$$\text{and} \quad \mathbb{E}_\varphi\left(F_{\text{down}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, g(\beta_{i+1})$$

define two areas (under two step-functions) both approximating the integral

$$F = \int_{\beta_n}^{\beta_0} g(\beta)\, \mathrm{d}\beta$$

(see Figure 5-1 for illustration). Using the same picture, we can visualise the difference between $\mathbb{E}_\varphi\left(F_{\text{down}}\right)$ and $\mathbb{E}_\varphi\left(F_{\text{up}}\right)$ by taking each block of the step-function areas and shading the part that is not overlapped by one of the other blocks (see Figure 5-2). By this, we obtain $(n-1)$ shaded squares (or rectangles) whose sum ("sum of squares") represents the desired difference

$$\begin{aligned} S &= \mathbb{E}_\varphi\left(F_{\text{down}}\right) - \mathbb{E}_\varphi\left(F_{\text{up}}\right) \\ &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, \left\{\mathbb{E}_\varphi\left[h(X_i')\right] - \mathbb{E}_\varphi\left[h(X_i)\right]\right\} \\ &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\, \left[g(\beta_{i+1}) - g(\beta_i)\right]. \end{aligned}$$

At this point, it is worth pointing out that approximating the area F can be a goal in itself. The area F corresponds to the log-ratio $\log\left[\frac{Z(\beta_n)}{Z(\beta_0)}\right]$ where $Z(\beta)$ denotes the normalisation constant of the distribution p_β, i.e.

$$Z(\beta) = \int \pi(x) \exp\left[-\beta\, h(x)\right]\, \mathrm{d}x.$$

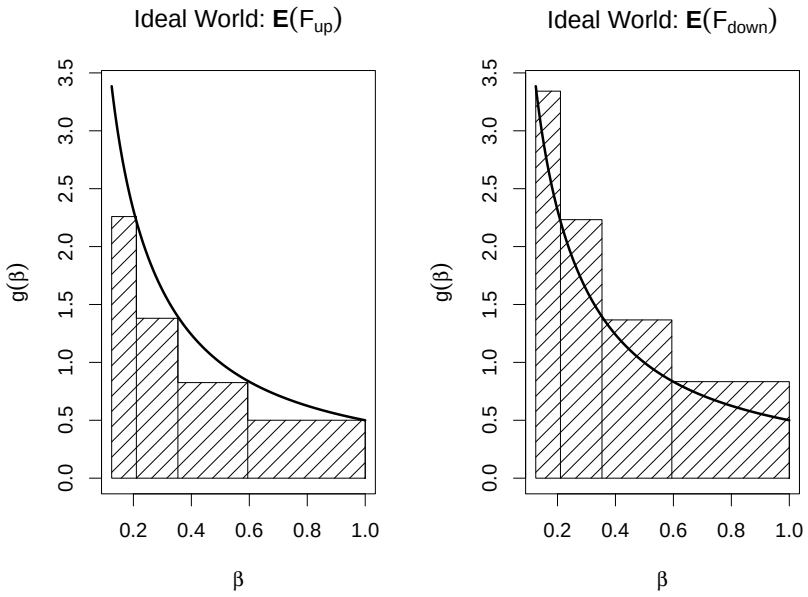

Figure 5-1: In the "ideal world" scenario, the anchor points of the step-functions defining the shaded $\mathbb{E}_{\varphi}(F_{\text{up}})$ and $\mathbb{E}_{\varphi}(F_{\text{down}})$ lie on the curve $g(\beta)$ because convergence is established immediately at each temperature.

To see this, let us derive the derivative of $Z(\beta)$ with respect to β:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\beta} Z(\beta) &= \int \pi(x) \frac{\mathrm{d}}{\mathrm{d}\beta} \exp\left[-\beta\, h(x)\right] \mathrm{d}x \\
&= Z(\beta) \int \frac{1}{Z(\beta)} \pi(x) \left[-h(x)\right] \exp\left[-\beta\, h(x)\right] \mathrm{d}x \\
&= -Z(\beta)\, \mathbb{E}_{p_\beta}\left[h(X)\right] \\
&= -Z(\beta)\, g(\beta)\,.
\end{aligned}$$

Rearranging this differential equation gives

$$g(\beta) = -\frac{1}{Z(\beta)} \left[\frac{\mathrm{d}}{\mathrm{d}\beta} Z(\beta)\right].$$

Integrating on both sides with respect to β in the limits β_n and β_0 finally yields the above feature

$$\begin{aligned}
F &= \int_{\beta_n}^{\beta_0} g(\beta)\, \mathrm{d}\beta \\
&= -\int_{\beta_n}^{\beta_0} \frac{1}{Z(\beta)} \left[\frac{\mathrm{d}}{\mathrm{d}\beta} Z(\beta)\right] \mathrm{d}\beta \\
&= -\log\left[Z(\beta)\right] \Big|_{\beta_n}^{\beta_0} \\
&= \log\left[\frac{Z(\beta_n)}{Z(\beta_0)}\right].
\end{aligned}$$

This feature allows us to estimate the normalisation constant $Z(\beta_0)$ of the target distribution if the normalisation constant $Z(\beta_n)$ of the hottest

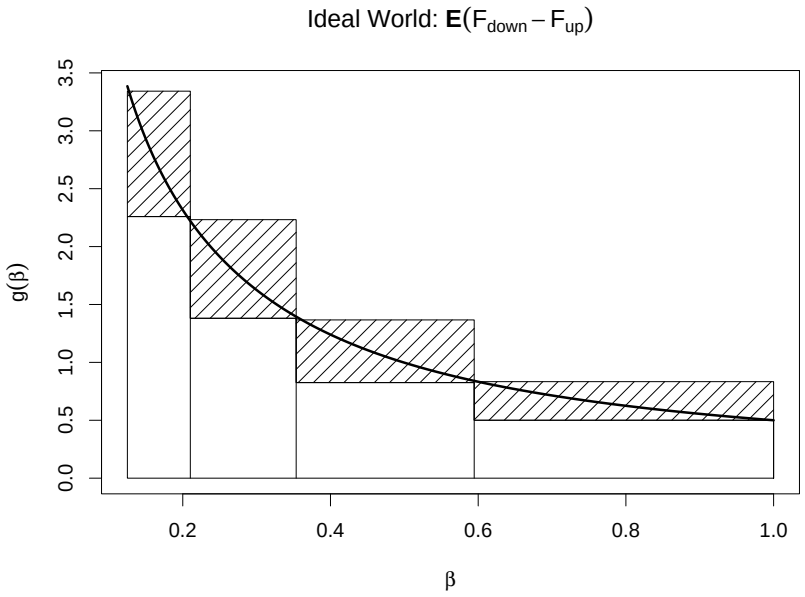

Figure 5-2: In the "ideal world" scenario, the anchor points defining the shaded "sum of squares" $S = \mathbb{E}_{\varphi}(F_{\text{down}}) - \mathbb{E}_{\varphi}(F_{\text{up}})$ lie on the curve $g(\beta)$ due to rapid convergence at each temperature. The term "sum of squares" is chosen because the shaded area representing S takes the form of several shaded squares (or rectangles) joint together.

distribution is known. This is for example possible if the normalisation constant of the prior distribution $\pi(x)$ is known and the hottest distribution is identical to the prior (at $\beta_n = 0$). We can use this property to calculate the marginal likelihood, which is a very important quantity in Bayesian model comparison. To demonstrate this, we will return for a moment to the Bayesian standard notation $p(\theta|y) \propto p(\theta)\, p(y|\theta)$ where $p(\theta)$ is the prior and $p(y|\theta)$ the likelihood. We will define the tempered version by $p_\beta(\theta|y) \propto p(\theta)\, p(y|\theta)^\beta$ so that $Z(\beta) = \int p(\theta)\, p(y|\theta)^\beta \,\mathrm{d}\theta$ and $g(\beta) = -\mathbb{E}_{p_\beta}\{\log[p(y|\theta)]\}$. When $\beta_n = 0$ and $\beta_0 = 1$, the quantity $(-F)$ equals the log-marginal-likelihood $\log[p(y)]$ of the target distribution, which can be derived as follows:

$$\begin{aligned}
-F &= -\log\left[\frac{Z(0)}{Z(1)}\right] \\
&= \log\left[\frac{Z(1)}{Z(0)}\right] \\
&= \log\left[\frac{\int p(\theta)\, p(y|\theta)\,\mathrm{d}\theta}{\int p(\theta)\,\mathrm{d}\theta}\right] \\
&= \log\left[\int p(\theta)\, p(y|\theta)\,\mathrm{d}\theta\right] \\
&= \log[p(y)]\,.
\end{aligned}$$

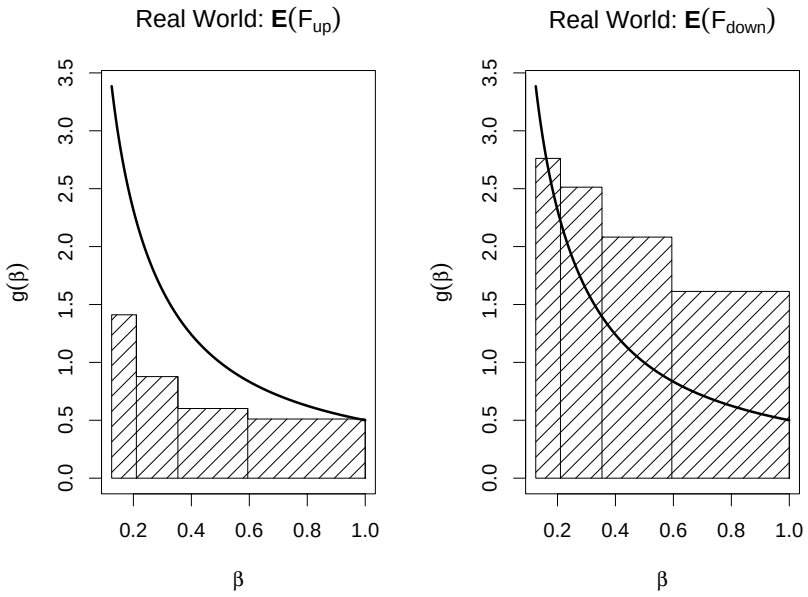

Figure 5-3: In the "real world" scenario, the anchor points of the step-functions defining the shaded $\mathbb{E}_\varphi(F_{\text{up}})$ and $\mathbb{E}_\varphi(F_{\text{down}})$ do not lie on the curve $g(\beta)$ due to slow convergence at each temperature.

For ways of approximating $p(y)$ and some applications, see for example Friel and Pettitt (2008).

Let us now return to our goal of optimising the temperatures in tempered transitions and thus to our notation $p_\beta(x) \propto \pi(x)\exp[-\beta\, h(x)]$ and $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$. We have discussed that the sum of squares $S = \mathbb{E}_\varphi(F_{\text{down}} - F_{\text{up}})$, which is a function of the inverse temperatures $\{\beta_i\}_{i=1}^n$, influences the acceptance probability

$$\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) = \min\{1, \exp[-(F_{\text{down}} - F_{\text{up}})]\}.$$

in tempered transitions. In the ideal world scenario (5.2), the approximations to the area F can be written as

$$\mathbb{E}_\varphi(F_{\text{up}}) = \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\,\mathbb{E}_{p_{\beta_i}}[h(X)]$$

$$\text{and} \quad \mathbb{E}_\varphi(F_{\text{down}}) = \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\,\mathbb{E}_{p_{\beta_{i+1}}}[h(X)]$$

and the sum of squares as

$$S = \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\;[g(\beta_{i+1}) - g(\beta_i)].$$

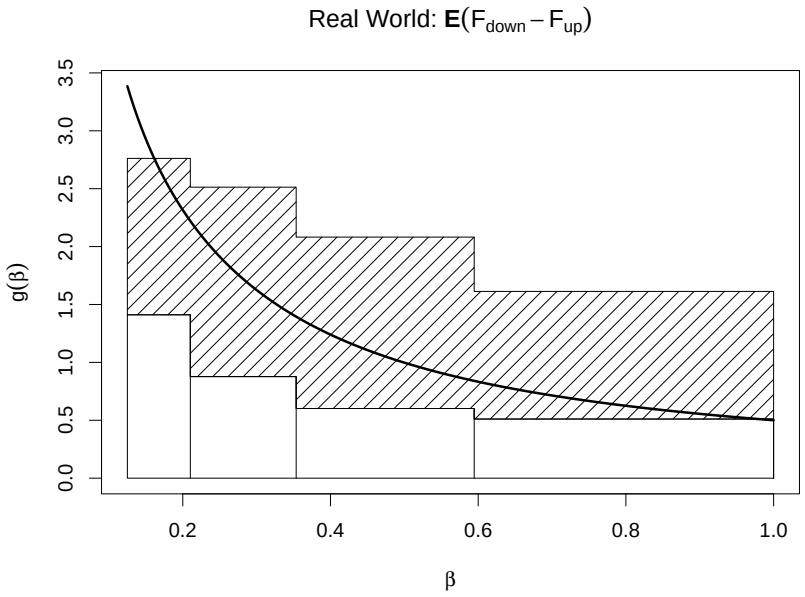

Figure 5-4: In the "real world" scenario, the anchor points defining the shaded "sum of squares" $S = \mathbb{E}_\varphi(F_{\text{down}}) - \mathbb{E}_\varphi(F_{\text{up}})$ do not lie on the curve $g(\beta)$ due to slow convergence at each temperature. The term "sum of squares" is chosen because the shaded area representing S takes the form of several shaded squares (or rectangles) joint together.

If the inverse temperatures satisfy the ordering constraint $\beta_{\min} = \beta_n < \ldots < \beta_1 = \beta_0$ as in the illustrations (Figures 5-1 and 5-2)), then increasing the number n of temperatures between $\beta_{\min}$ and β_0 leads to a smaller sum of squares $S = \mathbb{E}_\varphi(F_{\text{down}} - F_{\text{up}})$ because each of the step-function areas approximates the integral F better. As the expectation $\mathbb{E}_\varphi(F_{\text{down}} - F_{\text{up}})$ characterises the distribution $(F_{\text{down}} - F_{\text{up}})$ on which the acceptance probability

$$\alpha(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0) = \min\{1, \exp[-(F_{\text{down}} - F_{\text{up}})]\}$$

depends, it is intuitive that the smaller sum of squares also leads to a higher expected acceptance probability. This explains why increasing the number of temperatures improves the acceptance rate at least in the "ideal world" scenario (5.2). It remains to cover the "real world" scenario in which the Markov transitions kernels $\{T_{\beta_i}\}_{i=1}^n$ do not immediately establish convergence to the equilibrium distributions $\{p_{\beta_i}\}_{i=1}^n$ so that the joint distribution

$$\begin{aligned}&\varphi(x_0, \ldots, x_n, x'_{n-1}, \ldots, x'_0)\\ &\quad = p_{\beta_0}(x_0)\left[\prod_{i=0}^{n-1} T_{\beta_{i+1}}(x_i, x_{i+1})\right] T_{\beta_n}(x_n, x'_{n-1})\left[\prod_{i=0}^{n-2} T_{\beta_{i+1}}(x'_{i+1}, x'_i)\right]\end{aligned}$$

cannot be further simplified. Recall that we assume that the mixing is fast at the hottest temperature $\beta_{\min}$ so that tempered transitions mixes well overall. This means that we can assume that the initial state of the secondary chain X_0 has marginal distribution p_{β_0}, while the state X'_{n-1} generated at the hottest temperature has approximately marginal distribution $p_{\beta_n} = p_{\beta_{\min}}$. Since the remaining kernels are relatively slow in mixing, the auxiliary states x_i, $i = 1, \ldots, n-1$, generated in the heating-up process will be biased towards the coldest distribution, while the auxiliary states x'_i, $i = 0, \ldots, n-1$, generated in the cooling-down process will be biased towards the hottest distribution. As a result, the distributions of the energies $h(x_i)$, $i = 1, \ldots, n-1$, and $h(x'_i)$, $i = 0, \ldots, n-1$, will be biased towards the energy distribution at the coldest and hottest temperature, respectively. This implies that the anchor points $\mathbb{E}_\varphi[h(X_i)]$ and $\mathbb{E}_\varphi[h(X'_i)]$, $i = 1, \ldots, n-1$, will not lie on the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$, but somewhere near it. In Figure 5-3, the curve $g(\beta)$ decreases from $g(\beta_{\min})$ to $g(\beta_0)$ so that the anchor points $\mathbb{E}_\varphi[h(X_i)]$, $i = 1, \ldots, n-1$, defining

$$\mathbb{E}_\varphi(F_{\text{up}}) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \, \mathbb{E}_\varphi[h(X_i)]$$

lie below the curve $g(\beta)$ due to the bias towards $\mathbb{E}_\varphi[h(X_0)] = g(\beta_0)$, while the anchor points $\mathbb{E}_\varphi[h(X'_i)]$, $i = 1, \ldots, n-1$, defining

$$\mathbb{E}_\varphi(F_{\text{down}}) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \, \mathbb{E}_\varphi[h(X'_i)]$$

lie above the curve $g(\beta)$ due to the bias towards $\mathbb{E}_\varphi\left[h(X'_{n-1})\right] \approx g(\beta_{\min})$. The bias on both sides of the curve leads to a greater "sum of squares"

$$\begin{aligned} S &= \mathbb{E}_\varphi(F_{\text{down}}) - \mathbb{E}_\varphi(F_{\text{up}}) \\ &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \, \{\mathbb{E}_\varphi[h(X'_i)] - \mathbb{E}_\varphi[h(X_i)]\} \end{aligned}$$

than in the "ideal world" scenario (compare Figures 5-2 and 5-4). Again, increasing the number of temperatures between $\beta_{\min}$ and β_0 will lead to better approximations of the integral F and thus to a smaller sum of squares. In conclusion, raising the number of temperatures will most likely lead to higher acceptance rates in both the "ideal world" and the "real world" scenario.

In Neal's discussion, the difference between "real world" and "ideal world" is deduced to be due to the difference between fast mixing and slowly mixing transition kernels $\{T_{\beta_i}\}_{i=1}^n$. A point that Neal does not make is that we can also

achieve the "ideal world" scenario by running slowly mixing Markov transition kernels for several iterations at each temperature level. We can think of these iterations as a "burn-in" period because only the last state of this burn-in is relevant for the acceptance of the tempered transitions proposal due to some cancellation in the acceptance ratio. If the burn-in is long enough, the last auxiliary state at the current temperature level will be practically independent of the last state of the previous temperature level, which corresponds to the "ideal world" scenario. We can incorporate a burn-in period b at each temperature by running b iterations at each of the t distinct temperature levels. This means that we define the tempered transitions sampler by $n = tb$ inverse temperatures β_i, $i = 0, \ldots, n$, where

$$\beta_0 = \beta_1 = \cdots = \beta_b < \beta_{b+1} = \cdots = \beta_{2b} < \cdots < \beta_{(t-1)b+1} = \cdots = \beta_{tb}.$$

As many terms in the acceptance ratio cancel, we accept the path

$$(x_0, x_1, x_2, \ldots, x_{tb-2}, x_{tb-1}, x_{tb}, x'_{tb-1}, x'_{tb-2}, \ldots, x'_2, x'_1, x'_0)$$

with probability

$$\min\left\{1, \frac{\prod_{j=0}^{t-1} p_{\beta_{jb+1}}(x_{jb}) \prod_{k=0}^{t-1} p_{\beta_{kb}}(x'_{kb})}{\prod_{j=0}^{t-1} p_{\beta_{jb}}(x_{jb}) \prod_{k=0}^{t-1} p_{\beta_{kb+1}}(x'_{kb})}\right\}$$

where $p_{\beta_{jb}} \neq p_{\beta_{jb+1}}$ as the $\beta_b < \beta_{2b} < \cdots < \beta_{tb}$ mark the end of each distinct temperature level. Note that this probability agrees with accepting the "thinned" path

$$(x_0, x_b, x_{2b}, x_{3b}, \ldots, x_{(t-1)b}, x_{tb}, x'_{(t-1)b}, \ldots, x'_{2b}, x'_b, x'_0).$$

The method thus "pretends" to have used only the last iteration at each temperature.

To illustrate the possibility of achieving the "ideal world" scenario by incorporating burn-in periods, tempered transitions was applied to the toy problem of sampling from the mixture of normal distributions $\frac{1}{2}\,N(0,1) + \frac{1}{2}\,N(10,1)$. The tempered distributions were of the form $p_\beta(x) \propto \exp\left[-\beta\, h(x)\right]$ with energy function

$$h(x) = -\log\left\{\exp\left[-\frac{1}{2}\,x^2\right] + \exp\left[-\frac{1}{2}\,(x-10)^2\right]\right\}.$$

Figure 5-5 shows that the target distribution features two well separated modes which start merging together at inverse temperature $\beta_{\min} = \frac{1}{8}$. The $t = 5$ distinct temperature levels were set geometrically by $\beta_t = 8^{-(t-1)/4}$, $t = 1, \ldots, 5$

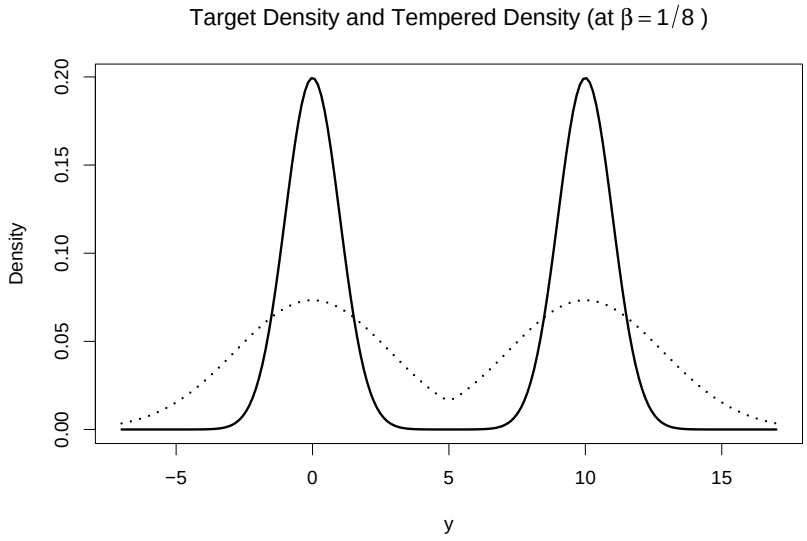

Figure 5-5: The target density $\frac{1}{2}N(0,1) + \frac{1}{2}N(10,1)$ features two well separated modes *(solid line)* which start merging together at inverse temperature $\beta = \frac{1}{8}$ *(dotted line)*.

and $\beta_0 = 1$ by definition. For simplicity, the same Metropolis transition kernel with proposal distribution $q(x, x') \sim N(x, 8^2)$ was used at each temperature level. To monitor the effect of the burn-in b, the states of the "thinned" secondary chain $(X_0, X_b, X_{2b}, X_{3b}, \ldots, X_{(t-1)b}, X_{tb}, X'_{(t-1)b}, \ldots, X'_{2b}, X'_b, X'_0)$ were stored at each iteration of the tempered transitions sampler. As the tempered transitions algorithm was run for $N = 10\,000$ iterations, there were also N samples of X_0, N samples of X_b, N samples of X_{2b} etc. It was therefore possible to estimate $\mathbb{E}_\varphi[h(X_{jb})]$, $j = 0, 1, \ldots, t$, by the empirical mean $\frac{1}{N}\sum_{k=1}^{N} h\left(X_{jb}^{(k)}\right)$. In a similar way, $\mathbb{E}_\varphi\left[h(X'_{jb})\right]$, $j = 0, 1, \ldots, t$, was estimated. These estimates were used to plot the step-function areas

$$\mathbb{E}_\varphi(F_{\text{up}}) = \sum_{j=0}^{t-1}(\beta_{jb} - \beta_{(j+1)b})\,\mathbb{E}_\varphi[h(X_{jb})]$$

$$\text{and} \quad \mathbb{E}_\varphi(F_{\text{down}}) = \sum_{j=0}^{t-1}(\beta_{jb} - \beta_{(j+1)b})\,\mathbb{E}_\varphi\left[h(X'_{jb})\right]$$

and the sum of squares

$$\begin{aligned} S &= \mathbb{E}_\varphi(F_{\text{down}}) - \mathbb{E}_\varphi(F_{\text{up}}) \\ &= \sum_{j=0}^{t-1}(\beta_{jb} - \beta_{(j+1)}b)\,\left\{\mathbb{E}_\varphi\left[h(X'_{jb})\right] - \mathbb{E}_\varphi[h(X_{jb})]\right\}. \end{aligned}$$

In this example, the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ was also plotted because its values could be obtained numerically. We have already seen the resulting plots for

this example (Figures 5-1 to 5-4) because they were used earlier to illustrate the difference between the "real world" and the "ideal world". To demonstrate the effect of slow convergence, the sampler was first run with $b = 1$. As we can see in Figures 5-3 and 5-4, the anchor points do not lie on the curve $g(\beta)$, which yields a greater sum of squares than in the ideal case. As discussed earlier, this behaviour is caused by the high dependencies between the states of the secondary chain. To show that a sufficiently long burn-in produces the "ideal case" scenario where the anchor points lie on the curve $g(\beta)$, tempered transitions was again run for $N = 10\,000$ iterations, but this time with a burn-in of $b = 3\,000$. As can be seen in Figures 5-1 and 5-2, this burn-in leads to the "ideal world" scenario with a smaller sum of squares. In this example, the smaller sum of squares raises the acceptance rate. It improves from 0.56 ("real world") to 0.65 ("ideal world"). This leads us to the question whether it is better to increase the burn-in b or the number t of distinct temperature levels. To investigate this point, the number of distinct temperature levels was kept constant by $t = 5$ with varying burn-in $b = 1, 2, 3, 4$ in one experiment (see Table 5-1), while $b = 1$ was fixed with different $t = 5, 10, 15, 20$ in another (see Table 5-2). This time, the integrated autocorrelation time $\tau(x)$ was monitored, too. It was approximated by Geyer's estimator (2.3) with respect to the theoretical mean $\mathbb{E}_{p_{\beta_0}}(X) = 5$ (rather than the empirical one) to obtain a higher accuracy. As can be seen in Table 5-1, both strategies raise the acceptance rate and diminish τ, which means that the mixing improves. As in both experiments the total number $n = tb$ (and thus the computational cost) grows in the same way ($n = 5, 10, 15, 20$), we can compare the quality of the improvement in the acceptance rate and in the mixing. We can see that, in both regards, placing more temperatures between $\beta_{\min}$ and β_0 is the better strategy. For completeness, it was also checked what happens if we keep the cost $n = 200$ constant, but change the way of distributing the cost $n = t \cdot b = 200 \cdot 1, 50 \cdot 4, 20 \cdot 10, 5 \cdot 40$ (see Table 5-3). Again, it was best to use up all resources for t. It is also interesting to see how the acceptance rate and the mixing change when $t = 2, 3, 4, \ldots, 10$ is slowly increased under fixed $b = 1$ (see Table 5-4). While the mixing becomes better as expected, the acceptance rate, at first thought surprisingly, decreases between $t = 2$ and $t = 4$ before adopting its usual behaviour of constant growth. The higher acceptance rates for small t can be explained by the fact that the proposal chain does not have many chances of moving away from the current state so that it is more likely to obtain acceptance probabilities of one at small t values than at the slightly

Increasing Burn-in at Temperature Levels

t	×	b	n	acceptance rate	$\hat{\tau}(x)$
5	×	1	5	0.556	2.95
5	×	2	10	0.571	2.58
5	×	3	15	0.584	2.36
5	×	4	20	0.603	2.34

Table 5-1: Increasing the burn-in b, while leaving the number t of distinct temperature levels constant improves the mixing.

Increasing the Number of Distinct Temperatures

t	×	b	n	acceptance rate	$\hat{\tau}(x)$
5	×	1	5	0.556	2.95
10	×	1	10	0.598	2.50
15	×	1	15	0.646	2.21
20	×	1	20	0.672	2.01

Table 5-2: Increasing the number t of distinct temperature levels, while leaving the burn-in b constant improves the mixing.

higher ones. This also explains why the autocorrelation time makes relatively big downward jumps between $t = 2$ and $t = 4$ and then descends at a slower pace. We can also decide on the most cost-efficient number of temperatures by comparing the increase in cost with the decrease in τ. In this example, this is $n = t \cdot b = 2 \cdot 1$.

Another point that Neal hardly explores is the possibility of optimising the spacing between temperatures for a fixed number of temperatures. Although he mentions that the spacing influences the acceptance rate and although he claims without proof that the geometric spacing is optimal in a toy example (see Section 5.4.1), he does not give any theoretical or practical advice on how to find an optimal spacing. We are therefore taking a novel approach by developing a tuning technique minimising the sum of squares. We are interested in it because a decrease in the sum of squares will hopefully induce the desired increase in the expected acceptance probability $\mathbb{E}_\varphi(\alpha)$. It seems quite probable that the tuning technique will lead to a temperature sequence giving a relatively high, albeit not maximal value of $\mathbb{E}_\varphi(\alpha)$. We cannot attain the maximal value because the optimisation problems are not equivalent due

Constant Cost

t	$\times$	b	n	acceptance rate	$\hat{\tau}(x)$
200	$\times$	1	200	0.877	1.32
50	$\times$	4	200	0.871	1.39
20	$\times$	10	200	0.832	1.45
5	$\times$	40	200	0.645	2.25

Table 5-3: It is better to use the available resources for the number t of distinct temperature levels than the burn-in b.

Slow Increase in Number of Distinct Temperatures

t	$\times$	b	n	acceptance rate	$\hat{\tau}(x)$
2	$\times$	1	2	0.600	5.55
3	$\times$	1	3	0.571	3.76
4	$\times$	1	4	0.553	3.16
5	$\times$	1	5	0.556	2.95
6	$\times$	1	6	0.558	2.91
7	$\times$	1	7	0.569	2.60
8	$\times$	1	8	0.587	2.48
9	$\times$	1	9	0.587	2.45
10	$\times$	1	10	0.588	2.46

Table 5-4: When increasing the number t of distinct temperature levels slowly, we observe that the acceptance rate first drops because there is a greater chance that the proposal is identical to the current state and therefore accepted with probability one when t is small. If t is further increased, the acceptance rate raises as expected.

to the fact that the expectation of a function is in general not equal to the function of the expectation, which means in our case that

$$\begin{aligned}&\mathbb{E}_\varphi\left[\alpha(X_0,\ldots,X_{n-1},X_n,X'_{n-1},\ldots,X'_0)\right]\\&\quad\neq\min\left\{1,\exp\left[-\mathbb{E}_\varphi\left(\sum_{i=0}^{n-1}(\beta_i-\beta_{i+1})\,\left[h(X'_i)-h(X_i)\right]\right)\right]\right\}.\end{aligned}$$

Since there is no simple connection between the sum of squares and the acceptance rate, we cannot predict one given the value of the other. This complicates the tuning of the number n of temperatures. Although we may be able to optimise the spacing for a given number of temperatures n (by minimising the sum of squares), we may not be able to determine an optimal number of temperatures n since we do not know which value $\mathbb{E}_\varphi\left\{\sum_{i=0}^{n-1}(\beta_i-\beta_{i+1})\,\left[h(X'_i)-h(X_i)\right]\right\}$ yields a cost-efficient acceptance rate

before running tempered transitions. It seems therefore best to find a reasonable number of temperatures by trial and error using preliminary runs. For example, we can find the optimal spacing for a given number of temperatures and then run tempered transitions based on this scheme. If the resulting acceptance rate is not satisfactory, we can choose a lower or higher number of temperatures as appropriate, optimise the sum of squares and then try again. Other criteria, such as the computational cost times the integrated autocorrelation time, may also be applied to find a cost-efficient scheme. In a similar manner, we could tune the hottest inverse temperature $\beta_{\min}$ (subject to being hot enough). Note, however, that a smaller $\beta_{\min}$ leads to a greater integral F that has to be approximated by $\mathbb{E}_\varphi(F_{\text{down}})$ and $\mathbb{E}_\varphi(F_{\text{up}})$ so that, intuitively, we also have to increase the number of temperatures to obtain the same sum of squares $S = [\mathbb{E}_\varphi(F_{\text{down}}) - \mathbb{E}_\varphi(F_{\text{up}})]$ as in the case of leaving $\beta_{\min}$ unchanged. In terms of cost-efficiency, it seems therefore not advisable to choose $\beta_{\min}$ much smaller than necessary.

The last point is that minimising the sum of squares

$$\mathbb{E}_\varphi\left\{\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\,[h(X_i') - h(X_i)]\right\}$$

in the "real world" scenario would involve minimising over all possible temperatures $\{p_{\beta_i}\}_{i=1}^n$ and transition kernels $\{T_{\beta_i}\}_{i=1}^n$ which we cannot tackle. However, as we will see, minimising the sum of squares

$$\begin{aligned} S(\{\beta_i\}_{i=0}^n) &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\,\{\mathbb{E}_\varphi[h(X_i')] - \mathbb{E}_\varphi[h(X_i)]\} \\ &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)] \end{aligned}$$

in the "ideal world" scenario (5.2) is possible since then we only have to optimise the temperatures $\{\beta_i\}_{i=1}^n$. Let us find out about the search space for the optimal set of temperatures in the next section.

5.3 Search space for optimal temperatures

5.3.1 Decreasing curve

For defining the search space, it helps to know that the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ is a decreasing function (or more precisely, a non-increasing function) on $(0, \infty)$ with derivative $g'(\beta) = -\text{var}_{p_\beta}[h(X)]$. We will also need this property later to

approximate the curve if it is not analytically available (Section 7.7.1).

We will show that the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ is non-increasing by verifying that its derivative $g'(\beta) = \frac{\mathrm{d}}{\mathrm{d}\beta}\mathbb{E}_{p_\beta}[h(X)]$ is non-positive. Let $Z(\beta)^{-1}$ denote the normalisation constant of the distribution $p_\beta(x) \propto \pi(x)\exp[-\beta\, h(x)]$ so that

$$Z(\beta) := \int \pi(x)\,\exp[-\beta\, h(x)]\,\mathrm{d}x\,.$$

To calculate $\frac{\mathrm{d}}{\mathrm{d}\beta}\mathbb{E}_{p_\beta}[h(X)]$, we will need the derivatives $\frac{\mathrm{d}}{\mathrm{d}\beta}p_\beta(x)$ and $\frac{\mathrm{d}}{\mathrm{d}\beta}Z(\beta)^{-1}$. The latter is given by

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{1}{Z(\beta)}\right] &= \frac{\mathrm{d}}{\mathrm{d}Z(\beta)}\left[\frac{1}{Z(\beta)}\right]\cdot\left[\frac{\mathrm{d}}{\mathrm{d}\beta}Z(\beta)\right] \qquad \text{(chain rule)}\\
&= -\frac{1}{Z(\beta)^2}\left[\frac{\mathrm{d}}{\mathrm{d}\beta}Z(\beta)\right]\\
&= -\frac{1}{Z(\beta)^2}\int \pi(x)\,\frac{\mathrm{d}}{\mathrm{d}\beta}\exp[-\beta\, h(x)]\,\mathrm{d}x \qquad (5.4)\\
&= -\frac{1}{Z(\beta)^2}\int \pi(x)\,[-h(x)]\exp[-\beta\, h(x)]\,\mathrm{d}x\\
&= \frac{1}{Z(\beta)}\mathbb{E}_{p_\beta}[h(X)]\,.
\end{aligned}$$

This derivative helps us deduce the former

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\beta}p_\beta(x) &= \pi(x)\,\frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{1}{Z(\beta)}\exp[-\beta\, h(x)]\right]\\
&= \pi(x)\left\{\left[\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{1}{Z(\beta)}\right]\exp[-\beta\, h(x)] + \frac{1}{Z(\beta)}\left[\frac{\mathrm{d}}{\mathrm{d}\beta}\exp[-\beta\, h(x)]\right]\right\} \qquad (5.5)\\
&\overset{(5.4)}{=} \pi(x)\left\{\frac{1}{Z(\beta)}\exp[-\beta\, h(x)]\,\mathbb{E}_{p_\beta}[h(X)] - \frac{1}{Z(\beta)}h(x)\exp[-\beta\, h(x)]\right\}\\
&= p_\beta(x)\left\{\mathbb{E}_{p_\beta}[h(X)] - h(x)\right\}.
\end{aligned}$$

It follows that the derivative of the curve $g(\beta)$ is non-positive:

$$\begin{aligned}
g'(\beta) &= \frac{\mathrm{d}}{\mathrm{d}\beta}\mathbb{E}_{p_\beta}[h(X)]\\
&= \frac{\mathrm{d}}{\mathrm{d}\beta}\int h(x)\,p_\beta(x)\,\mathrm{d}x\\
&= \int h(x)\,\frac{\mathrm{d}}{\mathrm{d}\beta}p_\beta(x)\,\mathrm{d}x\\
&\overset{(5.5)}{=} \int \left\{h(x)\,\mathbb{E}_{p_\beta}[h(X)] - [h(x)]^2\right\}p_\beta(x)\,\mathrm{d}x\\
&= \mathbb{E}_{p_\beta}[h(X)]\int h(x)\,p_\beta(x)\,\mathrm{d}x - \int [h(x)]^2\,p_\beta(x)\,\mathrm{d}x\\
&= \left\{\mathbb{E}_{p_\beta}[h(X)]\right\}^2 - \mathbb{E}_{p_\beta}\left\{[h(X)]^2\right\}\\
&= -\mathrm{var}_{p_\beta}[h(X)]\\
&\leq 0.
\end{aligned}$$

Hence, the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ is a decreasing function. Note that the decay at a particular β value depends on the variance $\mathrm{var}_{p_\beta}[h(X)]$: the greater this variance, the stronger the decay.

We can also derive the second derivative

$$\begin{aligned}
g''(\beta) &= \frac{\mathrm{d}^2}{\mathrm{d}\beta^2}\,\mathbb{E}_{p_\beta}[h(X)] \\
&= \frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{\mathrm{d}}{\mathrm{d}\beta}\,\mathbb{E}_{p_\beta}[h(X)]\right] \\
&= \frac{\mathrm{d}}{\mathrm{d}\beta}\left[-\mathrm{var}_{p_\beta}[h(X)]\right] \\
&= -\frac{\mathrm{d}}{\mathrm{d}\beta}\,\mathbb{E}_{p_\beta}\left\{\left\{h(X) - \mathbb{E}_{p_\beta}[h(X)]\right\}^2\right\} \\
&= -\frac{\mathrm{d}}{\mathrm{d}\beta}\int \left\{h(X) - \mathbb{E}_{p_\beta}[h(X)]\right\}^2 p_\beta(x)\,\mathrm{d}x \\
&= -\int \left\{h(X) - \mathbb{E}_{p_\beta}[h(X)]\right\}^2 \frac{\mathrm{d}}{\mathrm{d}\beta}\,p_\beta(x)\,\mathrm{d}x \\
&\overset{(5.5)}{=} -\int \left\{h(x) - \mathbb{E}_{p_\beta}[h(X)]\right\}^2 \left\{\mathbb{E}_{p_\beta}[h(X)] - h(x)\right\} p_\beta(x)\,\mathrm{d}x \\
&= \mathbb{E}_{p_\beta}\left\{\left\{h(X) - \mathbb{E}_{p_\beta}[h(X)]\right\}^3\right\}
\end{aligned}$$

which determines the shape of the curve $g(\beta)$. The shape may be convex, concave or of mixed behaviour. We will demonstrate in a toy example how the shape influences the spacing of the optimal temperatures (Section 6.4).

5.3.2 Ordering constraint for optimal temperatures

Due to the doubling of temperatures $\beta_1 = \beta_0$ in $\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$, the sum of squares can be simplified by

$$\begin{aligned}
S\left(\{\beta_i\}_{i=0}^n\right) &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)] \\
&= \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)] \\
&= S\left(\{\beta_i\}_{i=1}^n\right).
\end{aligned}$$

An equivalent optimisation problem is thus to find $\{\beta_i\}_{i=1}^n$ such that

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)]$$

is minimised. In the following, we will prove that the optimal temperatures satisfy the ordering constraint

$$\beta_{\min} = \beta_n < \cdots < \beta_1$$

where $\beta_1 = \beta_0$ is fixed by definition. This result is useful because it reduces the search space of the optimisation methods. First we will check that all optimal inverse temperatures lie between $\beta_{\min}$ and $\beta_1 = \beta_0$, then we will show that the optimal scheme is ordered. Both proofs are based on the previous result that the curve $g(\beta)$ is decreasing.

We will first consider including some inverse temperatures that are greater than the target temperature $\beta_1 = \beta_0$ in the temperature scheme. Such inverse temperatures would over-cool the target distribution and thus exacerbate its modes. Suppose the over-cooled temperatures would be put at the beginning of the temperature sequence. Then the secondary chain would start with over-cooling the target distribution which is not desirable but necessary for reversibility. The secondary chain would then go over to heating-up the over-cooled distributions until the hottest temperature is reached where mode swapping is possible. On the way back, the cooling-down process would again over-cool the target distribution so that the secondary chain hopefully visits the exaggerated peak of one of the target modes. In the last steps, the over-cooling would be reversed until the target temperature is reached. If the sampler does not move too far from the peak of the target mode, the target density at the final proposal should be fairly high. In brief, the hope is that over-cooling might increase the acceptance probability of tempered transitions. Unfortunately, this is not the case. In fact, over-cooling reduces the acceptance rate because it produces a larger sum of squares $\sum_{i=1}^{n-1}(\beta_i - \beta_{i+1})\ [g(\beta_{i+1}) - g(\beta_i)]$ than the one which would be obtained if the over-cooled temperatures were left out. To see this, let us remove the over-cooled temperatures one by one according to the following rule: if there is an index k such that $\beta_k = \max_i \{\beta_i\}$ and $\beta_k > 1$, then β_k is the coldest over-cooled inverse temperature and has to be taken out. Without loss of generality, let us assume that

$$\beta_{k+1} \leq \beta_{k-1} \leq \beta_k \,.$$

As the curve $g(\beta)$ is decreasing, we also have

$$g(\beta_{k+1}) \geq g(\beta_{k-1}) \geq g(\beta_k)$$

so that

$$(\beta_k - \beta_{k+1})\ [g(\beta_{k+1}) - g(\beta_k)] \geq (\beta_{k-1} - \beta_{k+1})\ [g(\beta_{k+1}) - g(\beta_{k-1})] \,. \tag{5.6}$$

It follows that the difference between the sum of squares obtained by the original scheme $\{\beta_i\}_i$ and the sum of squares obtained by the reduced scheme

$\{\beta_i\}_{i\neq k}$ is non-negative:

$$\begin{aligned}
&S\left(\{\beta_i\}_i\right) - S\left(\{\beta_i\}_{i\neq k}\right) \\
&\quad = (\beta_{k-1} - \beta_k)\,[g(\beta_k) - g(\beta_{k-1})] + (\beta_k - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_k)] \\
&\qquad - (\beta_{k-1} - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_{k-1})] \\
&\quad \overset{(5.6)}{\geq} (\beta_{k-1} - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_{k-1})] + (\beta_k - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_k)] \\
&\qquad - (\beta_{k-1} - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_{k-1})] \\
&\quad \geq (\beta_k - \beta_{k+1})\,[g(\beta_{k+1}) - g(\beta_k)] \\
&\quad \geq 0.
\end{aligned}$$

If we now consider the reduced scheme as the original scheme denoted by $\{\beta_i\}_i$, then we can repeat the procedure to improve the sum of squares until all the over-cooled temperatures have been discarded. This means that the optimal temperature scheme satisfies $\beta_i \leq \beta_0$ for all i. Similar ideas can be applied to show that $\beta_{\min} \leq \beta_i$ for all i.

We can also prove that the optimal inverse temperatures satisfy the ordering constraint $\beta_{\min} = \beta_n < \cdots < \beta_1$ by showing that any σ-permutation $\beta_{\sigma(1)}, \beta_{\sigma(2)}, \ldots, \beta_{\sigma(n)}$ of this ordered scheme will lead to a greater sum of squares than the strictly ordered version. Since the first inverse temperature has to be the target temperature, we will only consider σ-permutations satisfying $\sigma(1) = 1$ which ensures that we sample from the target distribution (at $\beta_{\sigma(1)} = \beta_1 = \beta_0$) in the tempered transitions algorithm. We will show by induction that ordering such a permuted sequence $\beta_{\sigma(1)}, \beta_{\sigma(2)}, \ldots, \beta_{\sigma(n)}$ will reduce the sum of squares. Let $\beta_{\rho(1)}, \beta_{\rho(2)}, \ldots, \beta_{\rho(n)}$ denote the strictly ordered sequence. The induction assumption is then

$$S\left(\left\{\beta_{\sigma(i)}\right\}_{i=1}^n\right) - S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^n\right) \geq 0. \tag{5.7}$$

Let us verify this assumption for $n = 3$. If $n = 3$, the only possible unordered permutation is $\beta_{\sigma(1)} = \beta_1$, $\beta_{\sigma(2)} = \beta_3$ and $\beta_{\sigma(3)} = \beta_2$. Ordering this permutation by setting $\rho(i) = i$ for $i = 1, 2, 3$, yields

$$\begin{aligned}
S\left(\left\{\beta_{\sigma(i)}\right\}_{i=1}^3\right) - S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^3\right) &= \underbrace{(\beta_1 - \beta_3)}_{>(\beta_1-\beta_2)}\,[g(\beta_3) - g(\beta_1)] - (\beta_1 - \beta_2)\,[g(\beta_2) - g(\beta_1)] \\
&\geq 0
\end{aligned}$$

so that the induction assumption is true for $n = 3$.

Now suppose that the assumption is true for $(n-1)$ temperatures. Then we can show that the assumption is also true for n temperatures. First let us

order the first $(n-1)$ temperatures $\beta_1 = \beta_{\sigma(1)}, \ldots, \beta_{\sigma(n-1)}$ by the permutation ρ satisfying $\beta_{\rho(n-1)} < \cdots < \beta_{\rho(1)} = \beta_1$. This implies that $\beta_{\rho(n-1)} \leq \beta_{\sigma(n-1)}$. As $g(\beta)$ is decreasing, this further implies that $g(\beta_{\rho(n-1)}) \geq g(\beta_{\sigma(n-1)})$ so that

$$\begin{aligned}
&(\beta_{\sigma(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\sigma(n-1)})\right] - (\beta_{\rho(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(n-1)})\right] \\
&\qquad \geq (\beta_{\rho(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\sigma(n)}) + g(\beta_{\rho(n-1)}) - g(\beta_{\sigma(n)})\right] \\
&\qquad \geq 0.
\end{aligned}$$

By applying this inequality and by assuming that the induction assumption (5.7) holds for $(n-1)$, we obtain

$$\begin{aligned}
&S\left(\left\{\beta_{\sigma(i)}\right\}_{i=1}^{n}\right) - S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}, \beta_{\sigma(n)}\right) \\
&\quad = S\left(\left\{\beta_{\sigma(i)}\right\}_{i=1}^{n-1}\right) + (\beta_{\sigma(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\sigma(n-1)})\right] \\
&\qquad - S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}\right) - (\beta_{\rho(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(n-1)})\right] \\
&\quad \geq (\beta_{\sigma(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\sigma(n-1)})\right] - (\beta_{\rho(n-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(n-1)})\right] \\
&\quad \geq 0.
\end{aligned}$$

This shows that the partially ordered sequence $\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}, \beta_{\sigma(n)}\right)$ does better than the unordered sequence $\left(\left\{\beta_{\sigma(i)}\right\}_{i=1}^{n}\right)$. If $\beta_{\sigma(n)} \leq \beta_{\rho(i)}$ for all i, then the sequence is actually in full order so that the proof is complete. Otherwise, it remains to show that a completely ordered sequence gives a smaller sum of squares than the partially ordered sequence. If the sequence is not yet in order, then there exists exactly one $k \in \{2, \ldots, n-1\}$, such that

$$\beta_{\rho(k)} < \beta_{\sigma(n)} < \beta_{\rho(k-1)} \qquad \text{and} \qquad g(\beta_{\rho(k)}) \geq g(\beta_{\sigma(n)}) \geq g(\beta_{\rho(k-1)}) \tag{5.8}$$

so that the following permutation $\tilde{\rho}$ defines a completely ordered sequence:

$$\begin{aligned}
\tilde{\rho}(i) &= \quad \rho(i) && \text{for } i = 1, \ldots, k-1, \\
\tilde{\rho}(k) &= \quad \sigma(n) && \text{and} \\
\tilde{\rho}(j) &= \rho(j-1) && \text{for } j = k+1, k+2, \ldots, n.
\end{aligned} \tag{5.9}$$

The sum of squares $S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}, \beta_{\sigma(n)}\right)$ of the partial ordering and the sum of squares $S\left(\left\{\beta_{\tilde{\rho}(i)}\right\}_{i=1}^{n}\right)$ of the complete ordering have many terms in common. If we write

$$\begin{aligned}
&S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}, \beta_{\sigma(n)}\right) \\
&\quad = S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{k-1}\right) + (\beta_{\rho(k-1)} - \beta_{\rho(k)}) \left[g(\beta_{\rho(k)}) - g(\beta_{\rho(k-1)})\right] \\
&\qquad + S\left(\left\{\beta_{\rho(j-1)}\right\}_{j=k+1}^{n}\right) + (\beta_{\sigma(n)} - \beta_{\rho(n-1)}) \left[g(\beta_{\rho(n-1)}) - g(\beta_{\sigma(n)})\right]
\end{aligned}$$

and

$$\begin{aligned}
S\left(\left\{\beta_{\tilde{\rho}(i)}\right\}_{i=1}^{n}\right) = S\left(\left\{\beta_{\tilde{\rho}(i)}\right\}_{i=1}^{k-1}\right) &+ (\beta_{\rho(k-1)} - \beta_{\sigma(n)}) \left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(k-1)})\right] \\
&+ (\beta_{\sigma(n)} - \beta_{\rho(k)}) \left[g(\beta_{\rho(k)}) - g(\beta_{\sigma(n)})\right] + S\left(\left\{\beta_{\tilde{\rho}(j)}\right\}_{j=k+1}^{n}\right),
\end{aligned}$$

then by (5.9) their difference reduces to the following positive term:

$$
\begin{aligned}
& S\left(\left\{\beta_{\rho(i)}\right\}_{i=1}^{n-1}, \beta_{\sigma(n)}\right) - S\left(\left\{\beta_{\tilde{\rho}(i)}\right\}_{i=1}^{n}\right) \\
&= \underbrace{\left(\beta_{\sigma(n)} - \beta_{\rho(n-1)}\right)}_{\geq (\beta_{\sigma(n)} - \beta_{\rho(k)}) \text{ by ordering of } \rho} \left[g(\beta_{\rho(n-1)}) - g(\beta_{\sigma(n)})\right] + \underbrace{\left(\beta_{\rho(k-1)} - \beta_{\rho(k)}\right)}_{> (\beta_{\rho(k-1)} - \beta_{\rho(n)}) \text{ by (5.8)}} \left[g(\beta_{\rho(k)}) - g(\beta_{\rho(k-1)})\right] \\
&\quad - \left(\beta_{\rho(k-1)} - \beta_{\sigma(n)}\right) \left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(k-1)})\right] - \left(\beta_{\sigma(n)} - \beta_{\rho(k)}\right) \left[g(\beta_{\rho(k)}) - g(\beta_{\sigma(n)})\right] \\
&\geq \underbrace{\left(\beta_{\sigma(n)} - \beta_{\rho(k)}\right)}_{> 0 \text{ by (5.8)}} \underbrace{\left[g(\beta_{\rho(n-1)}) - g(\beta_{\rho(k)})\right]}_{\geq 0 \text{ by ordering of } \rho} + \underbrace{\left(\beta_{\rho(k-1)} - \beta_{\rho(n)}\right)}_{> 0 \text{ by (5.8)}} \underbrace{\left[g(\beta_{\sigma(n)}) - g(\beta_{\rho(k-1)})\right]}_{\geq 0 \text{ by (5.8)}} \\
&\geq 0.
\end{aligned}
$$

In conclusion, the optimal temperature scheme satisfies the ordering constraint $\beta_{\min} = \beta_n < \cdots < \beta_1$ where $\beta_1 = \beta_0$ by definition. We will now investigate how to find optimal temperatures subject to this constraint.

5.4 Optimisation methods

5.4.1 Analytic optimisation

Toy study

In this section, we will discuss Neal's analytic toy example that he chooses to motivate the use of geometrically spaced temperatures in tempered transitions. We will follow here Neal's original argument which is based on the constraint $\beta_{\min} = \beta_n < \cdots < \beta_1 < \beta_0$ so that, in this section, geometric spacing means $\beta_i = \beta_{\min}^{i/n}$, $i = 0, \ldots, n$. As Neal's presentation is relatively short, we will fill in some additional steps.

Neal considers sampling from the standard multivariate normal distribution $X \sim N_d(0, I)$ by excursions over the tempered versions

$$p_\beta(x) \propto \exp\left[-\beta\, h(x)\right]$$

where the energy function $h(x)$ is defined by

$$h(x) := \frac{1}{2} \sum_{j=1}^{d} x_j^2.$$

The tempered distribution $p_\beta(x)$ is the multivariate normal distribution $N_d\left(0, \frac{1}{\beta}\, I\right)$. We can calculate the sum of squares in the "ideal world" scenario (5.2) in which the sum of squares $\mathbb{E}_\varphi\left(F_{\text{down}} - F_{\text{up}}\right)$ is given by

$$\mathbb{E}_\varphi\left(F_{\text{down}} - F_{\text{up}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left\{\mathbb{E}_{p_{\beta_{i+1}}}\left[h(X_i')\right] - \mathbb{E}_{p_{\beta_i}}\left[h(X_i)\right]\right\}.$$

To determine the sum, we need the energy mean $\mathbb{E}_{p_\beta}[h(X)]$ with respect to the tempered distribution $p_\beta(x)$. To derive this mean, let us write the components X_j, $j = 1, \ldots, d$, of $X \sim N_d\left(0, \frac{1}{\beta} I\right)$ by $X_j = Z_j/\sqrt{\beta}$ where $Z_j \sim N(0,1)$ so that the energy function is given by

$$h(X) = \frac{1}{2}\sum_{j=1}^{d} X_j^2 = \frac{1}{2\,\beta}\sum_{j=1}^{d} Z_j^2.$$

Recall that the sum of squared standard normal variables follows the chi-squared distribution $\sum_{j=1}^{d} Z_d^2 \sim \chi_d^2$ whose density will be denoted by ψ. Since this distribution has mean d, the energy mean $\mathbb{E}_{p_\beta}[h(X)]$ under the tempered distribution p_β is given by

$$\begin{aligned}\mathbb{E}_{p_\beta}[h(X)] &= \frac{1}{2\,\beta}\,\mathbb{E}_\psi\left(\sum_{j=1}^{d} Z_j^2\right)\\ &= \frac{d}{2\,\beta}\end{aligned}$$

so that the sum of squares becomes

$$\begin{aligned}\mathbb{E}_\varphi\left(F_{\text{down}} - F_{\text{up}}\right) &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\left\{\mathbb{E}_{p_{\beta_{i+1}}}[h(X_i')] - \mathbb{E}_{p_{\beta_i}}[h(X_i)]\right\}\\ &= \frac{d}{2}\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right).\end{aligned}$$

We can now minimise the sum of squares for a fixed hottest inverse temperature $\beta_{\min}$ and a fixed number n of temperatures subject to the ordering constraint $\beta_{\min} = \beta_n < \cdots < \beta_0$. Neal claims that, in this example, the optimal spacing of temperatures is geometric with $\beta_i = \beta_{\min}^{i/n}$, $i = 0, \ldots, n$, but neither proves the claim nor justifies it by an existing result. We will therefore verify this claim by induction. First, we need to check that the induction assumption holds for $n = 2$. If $n = 2$, then β_2 and β_0 are fixed with $\beta_0 = 1$, while β_1 can take any value in $(\beta_2, 1)$. In this case, we want to find β_1 that minimises the sum of squares

$$\begin{aligned}s(\beta_1) &= \sum_{i=0}^{1}(\beta_i - \beta_{i+1})\left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right)\\ &= (\beta_1 - \beta_2)\left(\frac{1}{\beta_2} - \frac{1}{\beta_1}\right) + (1 - \beta_1)\left(\frac{1}{\beta_1} - 1\right)\\ &= \left(\frac{\beta_2 + 1}{\beta_2}\right)\beta_1 + \frac{\beta_2 + 1}{\beta_1} + \text{constant}.\end{aligned}$$

Differentiating s with respect to β_1 gives

$$s'(\beta_1) = \frac{(\beta_2 + 1)}{\beta_2} - \frac{(\beta_2 + 1)}{\beta_1^2}.$$

Setting $s'(\beta_1) = 0$ implies $\beta_1^2 = \beta_2$. Since we are looking for an optimal solution β_1 satisfying $\beta_1^2 = \beta_2$ in the interval $(\beta_2, 1)$, the unique optimal solution is the geometric choice $\beta_1 = \beta_2^{1/2}$. The induction assumption is thus true for $n = 2$. It remains to show that the induction assumption holds for any n if it holds for $(n-1)$. This means that we are free to space $\beta_{n-1} \in (\beta_n, \beta_0)$. Once β_{n-1} is spaced, all other inverse temperatures are set geometrically by $\beta_i = \beta_{n-1}^{i/(n-1)}$, $i = 0, \ldots, n-2$, as this is then the optimal spacing based on the induction assumption. If we can show that $\beta_{n-1} = \beta_n^{(n-1)/n}$ minimises $\sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right)$, the induction will be complete. Spacing temperatures geometrically implies here that the squares (i.e. the components of the "sum of squares") are of the same size since, for $i = 0, \ldots, n-1$,

$$\begin{aligned}(\beta_i - \beta_{i+1})\left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right) &= \left(\beta_{n-1}^{i/(n-1)} - \beta_{n-1}^{(i+1)/(n-1)}\right)\left(\beta_{n-1}^{-(i+1)/(n-1)} - \beta_{n-1}^{-i/(n-1)}\right) \\ &= \left[\beta_{n-1}^{i/(n-1)}\left(1 - \beta_{n-1}^{1/(n-1)}\right)\right]\left[\beta_{n-1}^{-i/(n-1)}\left(\beta_{n-1}^{1/(n-1)} - 1\right)\right] \\ &= \left(1 - \beta_{n-1}^{1/(n-1)}\right)\left(\beta_{n-1}^{1/(n-1)} - 1\right).\end{aligned}$$

In consequence, the optimisation problem of setting β_{n-1} simplifies to minimising

$$\begin{aligned}s(\beta_{n-1}) &= \sum_{i=0}^{n-1}(\beta_i - \beta_{i+1})\left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right) \\ &= (\beta_{n-1} - \beta_n)\left(\beta_n^{-1} - \beta_{n-1}^{-1}\right) + (n-1)\left(1 - \beta_{n-1}^{1/(n-1)}\right)\left(\beta_{n-1}^{-1/(n-1)} - 1\right) \\ &= \beta_n^{-1}\,\beta_{n-1} + \beta_n\,\beta_{n-1}^{-1} + (n-1)\,\beta_{n-1}^{-1/(n-1)} + (n-1)\,\beta_{n-1}^{1/(n-1)} + \text{constant}.\end{aligned}$$

The derivative with respect to β_{n-1} is then

$$\begin{aligned}s'(\beta_{n-1}) &= \beta_n^{-1} - \beta_n\,\beta_{n-1}^{-2} - \beta_{n-1}^{-n/(n-1)} + \beta_{n-1}^{-(n-2)/(n-1)} \\ &= \beta_n^{-1} - \beta_n\,\beta_{n-1}^{-2} + \left(1 - \beta_{n-1}^{-2/(n-1)}\right)\beta_{n-1}^{-(n-2)/(n-1)}.\end{aligned}$$

Setting $\beta_{n-1} = \beta_n^{(n-1)/n}$ gives indeed $s'(\beta_{n-1}) = 0$. This choice is the unique solution of $s'(\beta_{n-1}) = 0$ because $s'(\beta_{n-1})$ is strictly increasing on $(0, 1)$, so that geometric spacing is here the unique optimal solution. To verify that $s'(\beta_{n-1})$ is strictly increasing on $(0, 1)$, we will show that $s'(\tilde{\beta}_{n-1}) - s'(\beta_{n-1}) > 0$ for

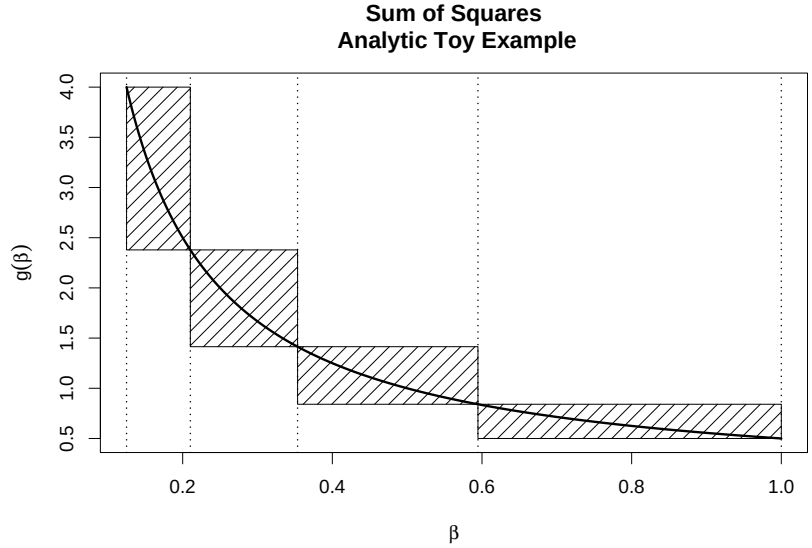

Figure 5-6: The figure shows the minimal sum of squares for $n = 5$ temperatures between $\beta_n = \frac{1}{8}$ and $\beta_1 = 1$ in the analytic toy problem where $g(\beta) = \frac{1}{2\beta}$.

$0 < \beta_{n-1} < \tilde{\beta}_{n-1} < 1$:

$$
\begin{aligned}
s'(\tilde{\beta}_{n-1}) - s'(\beta_{n-1}) = & \underbrace{\left(\beta_{n-1}^{-2} - \tilde{\beta}_{n-1}^{-2}\right)}_{>0} \beta_n \\
& + \underbrace{\left(1 - \tilde{\beta}_{n-1}^{-2/(n-1)}\right)}_{> \left(1 - \beta_{n-1}^{-2/(n-1)}\right)} \tilde{\beta}_{n-1}^{-(n-2)/(n-1)} \\
& - \left(1 - \beta_{n-1}^{-2/(n-1)}\right) \beta_{n-1}^{-(n-2)/(n-1)} \\
> & \underbrace{\left(1 - \beta_{n-1}^{-2/(n-1)}\right)}_{<0} \underbrace{\left(\tilde{\beta}_{n-1}^{-(n-2)/(n-1)} - \beta_{n-1}^{-(n-2)/(n-1)}\right)}_{<0} \\
> & 0.
\end{aligned}
$$

In conclusion, in this example, geometrically spaced temperatures minimise $\mathbb{E}_\varphi\left(F_{\text{down}} - F_{\text{up}}\right)$ for given n (see Figure 5-6).

It is interesting to see in this example how Neal decides on the number n of geometrically spaced temperatures which he denotes by $\beta_i = (1+\delta)^{-i}$, $i = 0, 1, \ldots, n$, where $\delta > 0$. In the new notation, each square in the sum of

squares is of size

$$\begin{aligned}(\beta_i - \beta_{i+1}) \left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right) &= \left(\frac{\beta_i}{\beta_{i+1}} - \frac{\beta_{i+1}}{\beta_{i+1}}\right) \left(\frac{\beta_{i+1}}{\beta_{i+1}} - \frac{\beta_{i+1}}{\beta_i}\right) \\ &= [(1+\delta) - 1] \left[1 - (1+\delta)^{-1}\right] \\ &= \delta \, \frac{\delta}{1+\delta} \\ &= \delta^2 \, \frac{1}{1+\delta}\end{aligned}$$

so that, for small δ, the mean $\mathbb{E}_\varphi \left(F_{\text{down}} - F_{\text{up}}\right)$ is approximately

$$\begin{aligned}\mathbb{E}_\varphi \left(F_{\text{down}} - F_{\text{up}}\right) &= \frac{d}{2} \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left(\frac{1}{\beta_{i+1}} - \frac{1}{\beta_i}\right) \\ &= \frac{d}{2} \, n \, \delta^2 \, \frac{1}{1+\delta} \\ &\approx \frac{d}{2} \, n \, \delta^2.\end{aligned}$$

As we want to reach a fixed minimal inverse temperature $\beta_{\min} = \beta_n = (1+\delta)^{-n}$ by the geometric scheme, we need to find a reasonable relationship between the number of temperatures $n(\delta, \beta_{\min})$, the spacing δ and the minimal inverse temperature $\beta_{\min}$. Neal suggests $n(\delta, \beta_{\min}) \approx -\frac{1}{\delta} \log(\beta_{\min})$, which might be motivated by $\mathbb{E}_\varphi \left(F_{\text{down}}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \frac{1}{\beta_{i+1}}$ being an approximation of the integral $F = \int_{\beta_{\min}}^1 \frac{1}{\beta} \, \mathrm{d}\beta = -\log(\beta_{\min})$ which implies that

$$\begin{aligned}-\log(\beta_{\min}) &\approx \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \, \frac{1}{\beta_{i+1}} \\ &= n \, \delta.\end{aligned}$$

Rearranging this approximation gives Neal's choice $n(\delta, \beta_{\min}) \approx -\frac{1}{\delta} \log(\beta_{\min})$. Moreover, Neal suggests setting $\delta \approx -1/\left[d \, \log(\beta_{\min})\right]$, where d is the dimension of the target distribution, so that

$$n(\beta_{\min}) \approx d \, \left[\log(\beta_{\min})\right]^2.$$

Neal justifies this choice by reckoning that the resulting mean and variance of the random variable $(F_{\text{down}} - F_{\text{up}})$ lead to a "reasonable" acceptance rate. To see what is meant by "reasonable", let us derive the variance of $(F_{\text{down}} - F_{\text{up}})$. we need again that the sum of squared standard normal variables follows the chi-squared distribution $\sum_{j=1}^d Z_d^2 \sim \chi_d^2$ with variance $2d$, in which case the energy $h(X)$ has variance

$$\begin{aligned}\operatorname{var}_{p_\beta} \left[h(X)\right] &= \left(\frac{1}{2\,\beta}\right)^2 \operatorname{var}_\psi \left(\sum_{j=1}^d Z_j^2\right) \\ &= \frac{d}{2\,\beta^2}\end{aligned}$$

under the tempered distribution $p_\beta(x)$. For geometric temperatures, the variance $\mathrm{var}_\varphi\left(F_{\mathrm{down}} - F_{\mathrm{up}}\right)$ is then given by

$$\begin{aligned}
\mathrm{var}_\varphi\left(F_{\mathrm{down}} - F_{\mathrm{up}}\right) &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})^2 \,\mathrm{var}_{p_{\beta_{i+1}}}\left[h(X)\right] + \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})^2 \,\mathrm{var}_{p_{\beta_i}}\left[h(X)\right] \\
&= \frac{d}{2} \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1})^2 \left(\frac{1}{\beta_{i+1}^2} + \frac{1}{\beta_i^2}\right) \\
&= \frac{d}{2} \sum_{i=0}^{n-1} \left[(1+\delta) - 1\right]^2 \left(1 + \frac{1}{(1+\delta)^2}\right) \\
&\approx d\, n\, \delta^2 \,.
\end{aligned}$$

If we plug the recommended $\delta \approx -1/\left[d \,\log\left(\beta_{\mathrm{min}}\right)\right]$ into the mean and variance formulae, we obtain

$$\begin{aligned}
\mathbb{E}_\varphi\left(F_{\mathrm{down}} - F_{\mathrm{up}}\right) &\approx \frac{d}{2}\, n\, \delta^2 \\
&\approx \frac{1}{2} \\
\text{and} \qquad \mathrm{var}_\varphi\left(F_{\mathrm{down}} - F_{\mathrm{up}}\right) &\approx d\, n\, \delta^2 \\
&\approx 1
\end{aligned}$$

which are meant to ensure reasonable acceptance probabilities

$$\alpha = \min\left\{1, \exp\left[-\left(F_{\mathrm{down}} - F_{\mathrm{up}}\right)\right]\right\}$$

in the tempered transitions algorithm. As we cannot deduce the value of the expected acceptance probability from a given value for the sum of squares, we cannot discuss Neal's recommendation any further.

General solution

Having seen that it is possible to minimise the sum of squares analytically, we will try to find a general analytic solution to the optimisation problem. Let us return to the ordering constraint

$$\beta_{\mathrm{min}} = \beta_n < \cdots < \beta_1 = \beta_0$$

so that we are looking for $\{\beta_i\}_{i=1}^n$ giving the lowest sum of squares

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1}) \,\left[g(\beta_{i+1}) - g(\beta_i)\right].$$

We will try to find a general analytic solution to the optimisation problem in the simplified problem of optimising $n = 3$ temperatures. Without loss of generality, we will assume that $\beta_1 = 1$, $\beta_3 = 0$, $g(\beta_1) = 0$ and $g(\beta_3) = 1$.

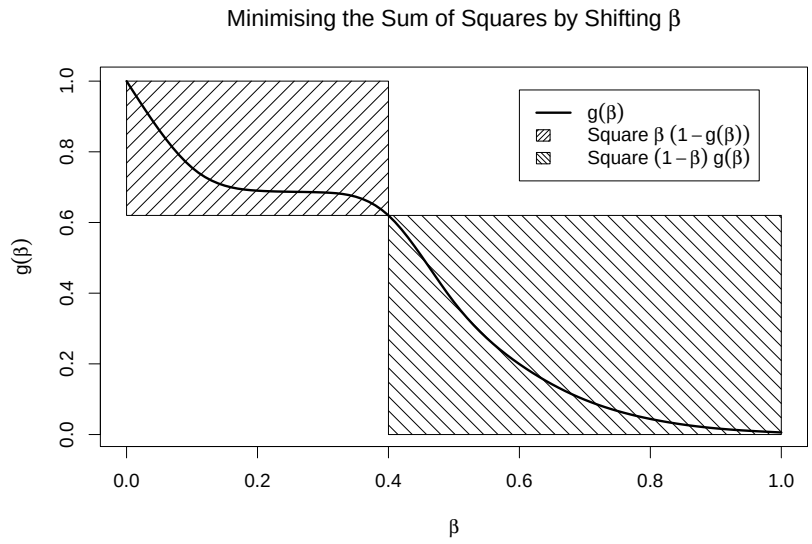

Figure 5-7: For illustration, let us assume that the decreasing curve $g(\beta)$ between $\beta_3 = 0$ and $\beta_1 = 1$ satisfies $g(\beta_3) = 1$ and $g(\beta_1) = 0$. The optimisation problem in its simplest form is then to find only one inverse temperature β between $\beta_1 = 1$ and $\beta_3 = 0$ such that the sum of squares $s(\beta) = \beta\,[1 - g(\beta)] + (1-\beta)\,g(\beta)$ is minimised. To distinguish the squares $\beta\,[1 - g(\beta)]$ and $(1 - \beta)\,g(\beta)$, two different types of shading are used in the illustration.

We can also drop the index of β_2 so that $\beta_2 = \beta$ and $g(\beta_2) = g(\beta)$. The optimisation problem is thus to find β that minimises the sum of squares

$$s(\beta) = \beta\,[1 - g(\beta)] + (1 - \beta)\,g(\beta) \tag{5.10}$$

where $g(\beta)$ is a decreasing function on $[0, 1]$ (see Figure 5-7 for illustration). Setting the derivative

$$s'(\beta) = 1 - 2\,g(\beta) + (1 - 2\,\beta)\,g'(\beta)$$

equal to zero yields the optimal solution

$$\beta^* = \frac{1 - 2\,g(\beta^*) + g'(\beta^*)}{2\,g'(\beta^*)}\,.$$

We can see that this solution is not easy to determine analytically. As the optimisation is even harder when $n > 3$, it seems best to minimise the sum of squares by deterministic or stochastic optimisation methods. We can speed up these methods by implementing the ordering constraint on the optimal temperatures. It would help if we could simplify the search further. In Neal's

toy example discussed in the previous section, the optimal temperature scheme leads to equally sized squares

$$(\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)] = (\beta_j - \beta_{j+1})\,[g(\beta_{j+1}) - g(\beta_j)]\,, \qquad i \neq j,$$

so that the question occurs whether equally sized squares imply the optimality of the underlying temperature scheme. Unfortunately, this is not the case as a simple counter-example shows. Consider the above simplified optimisation problem (5.10) with $g(\beta) := 1 - \beta^2$. Solving this problem yields the optimal inverse temperature $\beta^* = (1+\sqrt{7})/6$. If we calculate the size of the two squares defined by the optimal solution, we will find that the first square is of size $\beta^*\,(1-g(\beta^*)) = 0.22$, while the second square is of size $(1-\beta^*)\,g(\beta^*) = 0.25$ and thus greater. Hence, the equal size of squares is not necessarily a feature of the optimal solution so that we cannot simplify the optimisation problem further. Let us therefore move on to discussing two possible numerical optimisation methods, simulated annealing and dynamic programming.

5.4.2 Simulated annealing

Algorithm

Simulated annealing is a stochastic optimisation method (Kirkpatrick et al. 1983, Geman and Geman 1984) in which MCMC steps are carried out with respect to a steadily cooling tempered version of the target distribution (which represents the optimisation problem) so that eventually all the mass of the target distribution contracts to the global optimum. If we want to use simulated annealing to find the optimal inverse temperatures that minimise the sum of squares

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)]$$

subject to the constraint $\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$, then we will usually set up the method with respect to the equivalent problem of maximising

$$\exp\left\{-S\left(\{\beta_i\}_{i=1}^n\right)\right\}.$$

This equivalent problem can be considered the target distribution of simulated annealing. For optimisation, we will need tempered versions of this target distribution. We can define these tempered distributions by the temperature parameter T (or equivalently by the inverse temperature $\frac{1}{T}$) by setting

$$\psi_T\left(\{\beta_i\}_{i=1}^n\right) \propto \exp\left\{-\frac{1}{T}\,S\left(\{\beta_i\}_{i=1}^n\right)\right\}.$$

In the following, we will use the temperature parameter T to refer to a tempered distribution. By definition, the target temperature is $T = 1$. If T is greater than one, we obtain a hotter distribution. Otherwise if T is smaller than one, we obtain a colder distribution. In simulated annealing, the temperature is slowly lowered to exaggerate the target modes more and more. As the temperature T goes to zero, all the mass of the tempered distribution is concentrated at the global maxima of the target density. To find the global maxima, MCMC steps are run with respect to the tempered distribution ψ_{T_k} at each $k = 0, 1, 2, \ldots$ As k goes to infinity, the temperature T_k tends to zero. In practice, a finite decreasing sequence of temperatures $\{T_k\}_{k \in \underline{m}}$ where

$$\underline{m} := \{1, 2, \ldots, m\}$$

is used with T_m being a very small positive temperature. The simulated annealing algorithm is then:

Algorithm 5.1:

Step 1 Start at $\left\{\beta_i^{(0)}\right\}_{i=1}^n$.

Step 2 For $k = 1, 2, \ldots, m$, generate $\left\{\beta_i^{(k)}\right\}_{i=1}^n$ by an MCMC kernel which is reversible with respect to ψ_{T_k}.

Step 3 Return $\left\{\beta_i^{(m)}\right\}_{i=1}^n$.

Note that, although MCMC steps are carried out, simulated annealing is not an MCMC method because the steady change in temperature prohibits convergence to any of the tempered distributions ψ_{T_k}, $k \in \underline{m}$, so that its states cannot be considered samples from any distribution. Anyway, sampling is not the aim of simulated annealing; its object is optimisation.

Convergence

It is known that simulated annealing converges to the global optimum under certain theoretical conditions. Unfortunately, these conditions cannot be verified in practice so that convergence needs to be checked on a case-by-case basis with the risk that convergence is falsely diagnosed. In practice, the time of convergence also depends on the temperature scheme ("annealing schedule") used in simulated annealing.

Suppose in general that $S(\theta)$ is a cost function on a finite state space and that simulated annealing is run with respect to the tempered distribution $\psi_{T_k}(\theta) \propto \exp\left[-\frac{1}{T_k} S(\theta)\right]$ where $T_k \to 0$ as $k \to \infty$. From theory, we know that simulated annealing converges in probability to the global minimum of the cost function $S(\theta)$ if and only if $\sum_{k=1}^{\infty} \exp\left(-d^*/T_k\right) = +\infty$ where d^* is the maximum "depth" of all states which are local but not global minima (Hajek 1988). In particular, if the annealing scheme is logarithmic of the form $T_k = c/\log(1+k)$, $k \in \underline{m}$, then simulated annealing converges if and only if $c \geq d^*$. Unfortunately, the maximum "depth" d^* is virtually impossible to determine in practice so that this convergence result cannot be verified in practice. Since in addition the concept of the "depth" of a local minimum is complicated, we will here omit its definition; it can be found in Hajek (1988) or in Robert and Casella (1999, Section 5.2.3). For logarithmic annealing schemes $T_k = c/\log(1+k)$, $k \in \underline{m}$, Geman and Geman (1984) derive another bound for the value of the constant c. Unfortunately, this bound is substantially larger than d^* (Hajek 1988). Similarly, Geman and Geman (1984) report that their choice is far too large to be of practical value so that they define the logarithmic annealing scheme $T_k = c'/\log(1+k)$, $k \in \underline{m}$, for their experiments by some small constant c' which proves to be satisfactory. Note that the logarithmic annealing scheme is not the only possible scheme. Some practical guidance on the choice of the annealing scheme can be found in Stander and Silverman (1994) where families of linear, geometric, reciprocal and logarithmic annealing schemes are explored in some image analysis examples. In these examples, all the schemes achieve satisfactory results although the logarithmic scheme performs best.

Implementation

We will here suggest two ways of implementing simulated annealing for finding the optimal set of inverse temperatures $\{\beta_i\}$ minimising the sum of squares

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i)\right].$$

These two ways will be tested later in Chapter 6. One implementation performs a search over the unconstrained search space $\{\beta_i\}_{i=1}^n \in \{\beta_0\} \times [\beta_{\min}, \beta_0]^{n-2} \times \{\beta_{\min}\}$. The unconstrained search space is deliberately chosen because it gives us extra confidence in detecting convergence of the algorithm since the search over the unconstrained space should reduce the risk of several runs getting stuck in the same local optimum if the optimisation problem is multimodal. The other implementation only considers temperature schemes of the form

$\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$ and is thus faster. Both versions will be based on the logarithmic annealing scheme

$$T_k = \frac{T_{\max} T_{\min} \left[\log(m+2) - \log(2)\right]}{T_{\min} \log(m+2) - T_{\max} \log(2) + (T_{\max} - T_{\min}) \log(k+2)}, \quad k \in \underline{m},$$

as defined in Stander and Silverman (1994). To avoid confusion, we will use "temperatures" to refer to the temperatures $\{T_k\}_{k \in \underline{m}}$ of the annealing scheme, while we will use "inverse temperatures" to refer to the inverse temperatures $\{\beta_i\}_{i=1}^n$ of the tempered transitions algorithm which are to be optimised by simulated annealing.

The first simulated annealing version is defined on the unconstrained search space $\{\beta_i\}_{i=1}^n \in \{\beta_0\} \times [\beta_{\min}, \beta_0]^{n-2} \times \{\beta_{\min}\}$ so that the initial set of inverse temperatures is drawn uniformly by $\beta_i \sim U(\beta_{\min}, \beta_0)$ for $i = 2, \ldots, n-1$. Note that $\beta_1 = \beta_0$ and $\beta_n = \beta_{\min}$ are fixed by definition. At each temperature T_k, $k \in \underline{m}$, the algorithm updates the inverse temperatures β_i, $i = 2, \ldots, n-1$, component-wise in a complete sweep before moving on to the next temperature. The component-wise update uses a normal proposal with reflection at the boundaries of the interval $[\beta_{\min}, \beta_0]$. Such a proposal increases the efficiency of the update because it diminishes the probability of a proposal lying outside the interval, which would always be rejected due to the constraint $\beta_i \in [\beta_{\min}, \beta_0]$ for all i. We will illustrate how the reflected proposal mechanism works in Figures 5-8 to 5-10. Suppose β_i is the value of the current component i. Then we draw $z \sim N(0, \sigma_T^2)$ (where the step size σ_T may depend on the current temperature value T). If $(\beta_i + z)$ lies inside the interval (Figure 5-8), we will use this value for the proposal $\beta_i' = (\beta_i + z)$. If $(\beta_i + z)$ lies outside the interval, then we will reflect this value at the closest barrier, either at $\beta_{\min}$ or at β_0. That means that $(\beta_i + z) < \beta_{\min}$ will be reflected at $\beta_{\min}$ by $\beta_i^* = [2\,\beta_{\min} - (\beta_i + z)]$, while $(\beta_i + z) > \beta_0$ will be reflected at β_0 by $\beta_i^* = [2\,\beta_0 - (\beta_i + z)]$. Ideally, the reflected value lies in the interval $[\beta_{\min}, \beta_0]$ (Figure 5-9), in which case we will use it as the proposal β_i' and accept it with the usual Metropolis-Hastings acceptance probability. It may however happen that the reflected value is cast over the interval into the other zero-constraint area (Figure 5-10) so that we cannot use the reflected value as proposal. In this case, it is legitimate to start all over again and to continue drawing and reflecting new z values as described above until the procedure produces a value that lies in the interval and can therefore be used as a proposal. The generated proposal follows the proposal

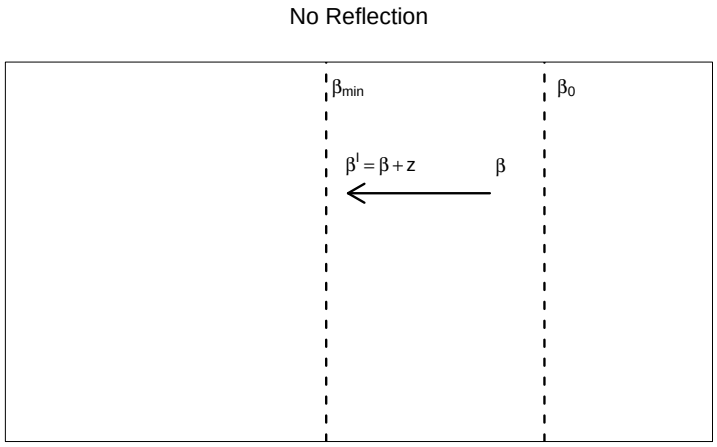

Figure 5-8: If the normal proposal $\beta' = (\beta + z)$ centred at β lands inside the interval $[\beta_{\min}, \beta_0]$, then it will be used as proposal state in the MCMC step.

distribution

$$q(\beta_i, \beta_i') = q_Z(\beta_i - \beta_i') + q_Z(2\beta_{\min} - \beta_i' - \beta_i) + q_Z(2\beta_0 - \beta_i' - \beta_i) \qquad \beta_i, \beta_i' \in [\beta_{\min}, \beta_0]$$

where $q_Z(\cdot)$ denotes the density of $N(0, \sigma_T^2)$. To understand the form of the proposal distribution, recall that the proposal mechanism does not stop until it produces a proposal in the interval so that we already know that $\beta_i' \in [\beta_{\min}, \beta_0]$. The only uncertainty is whether the proposal is generated by reflection, and if so, at which barrier. The first possibility is that no reflection takes place, which is equivalent to the case that the random variable is $z = (\beta_i - \beta_i')$. The second possibility is that the reflection is centred at $\beta_{\min}$ so that the random variable must be $z = (2\beta_{\min} - \beta_i' - \beta_i)$. The last possibility is that the reflection occurs at β_0, in which case $z = (2\beta_0 - \beta_i' - \beta_i)$. To account for all these possibilities, we add all the three possible density values of z, namely $q_Z(\beta_i - \beta_i')$, $q_Z(2\beta_{\min} - \beta_i' - \beta_i)$ and $q_Z(2\beta_0 - \beta_i' - \beta_i)$, together. We know that the density q_Z of $N(0, \sigma_T^2)$ satisfies $q_Z(z) = q_Z(-z)$. In consequence, the proposal distribution

$$\begin{aligned} q(\beta_i, \beta_i') &= q_Z(\beta_i - \beta_i') + q_Z(2\beta_{\min} - \beta_i' - \beta_i) + q_Z(2\beta_0 - \beta_i' - \beta_i) \\ &= q_Z(-(\beta_i - \beta_i')) + q_Z(2\beta_{\min} - \beta_i' - \beta_i) + q_Z(2\beta_0 - \beta_i' - \beta_i) \\ &= q_Z(\beta_i' - \beta_i) + q_Z(2\beta_{\min} - \beta_i - \beta_i') + q_Z(2\beta_0 - \beta_i - \beta_i') \\ &= q(\beta_i', \beta_i) \end{aligned}$$

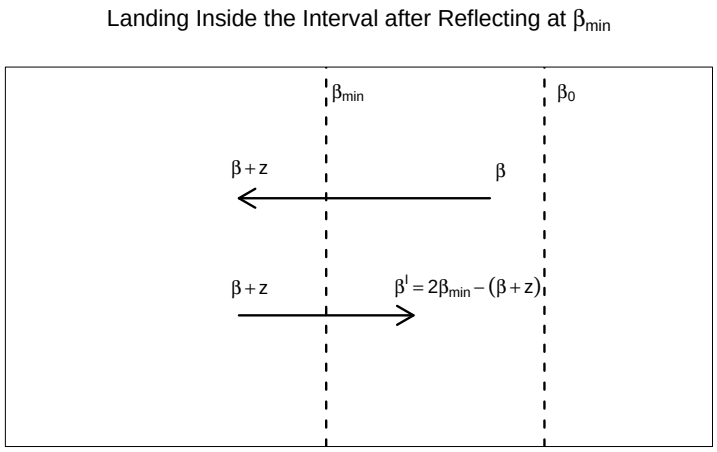

Figure 5-9: If the normal proposal $(\beta + z)$ centred at β lands outside the interval $[\beta_{\min}, \beta_0]$, then it is reflected at the nearest interval barrier, here $\beta_{\min}$. If the reflected proposal $\beta' = [2\,\beta_{\min} - (\beta + z)]$ lands inside the interval, it will be used as proposal state in the MCMC step.

is symmetric. The pseudo-code for the component-wise update at temperature T can be described as follows:

Algorithm 5.2:

Step 1 Draw $z \sim N(0, \sigma_T^2)$.

Step 2 Set

$$\beta_i^* = \begin{cases} \beta_i + z & \text{if } (\beta_i + z) \in [\beta_{\min}, \beta_0], \\ 2\,\beta_{\min} - (\beta_i + z) & \text{if } (\beta_i + z) < \beta_{\min}, \\ 2\,\beta_0 - (\beta_i + z) & \text{if } (\beta_i + z) > \beta_0. \end{cases}$$

Step 3 If $\beta_i^* \in [\beta_{\min}, \beta_0]$, then set the proposal state $\beta_i' = \beta_i^*$. Otherwise go to Step 1.

Step 4 As this proposal is symmetric, accept β_i' with probability

$$\alpha(\beta_i, \beta_i') = \min\left\{1, \frac{\psi_T\left(\beta_i' \middle| \{\beta_k\}_{k \in \underline{n} \setminus \{i\}}\right)}{\psi_T\left(\beta_i \middle| \{\beta_k\}_{k \in \underline{n} \setminus \{i\}}\right)}\right\}.$$

As the state space $[\beta_{\min}, \beta_0]$ for each temperature is bounded and relatively small, the step size σ_T can be set constant for all temperatures. We will later

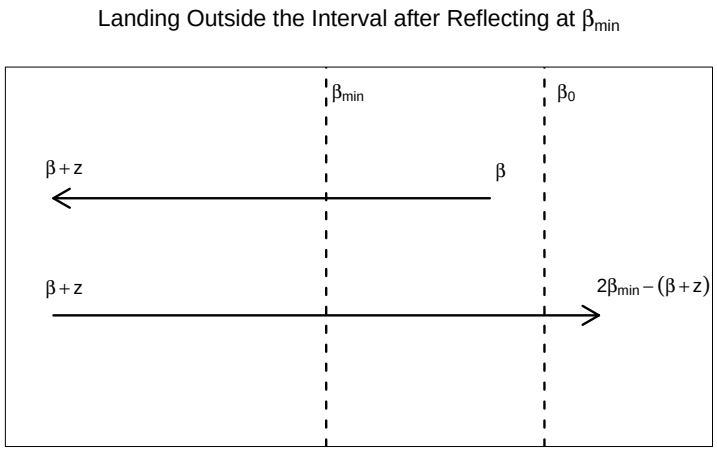

Figure 5-10: If the normal proposal $(\beta + z)$ centred at β lands outside the interval $[\beta_{\min}, \beta_0]$, then it is reflected at the nearest interval barrier, here $\beta_{\min}$. If the reflected proposal $[2\,\beta_{\min} - (\beta + z)]$ lands outside the interval, it will be discarded straightaway and a new normal proposal centred at β will be drawn.

use $\sigma_T = 10^{-3}$ because it gives a sufficient accuracy (Section 6.3). When experimenting with different choices of σ_T, one will find that, at some point, it is not worth using a smaller σ_T than the current choice because at this point the resulting higher precision in the optimal inverse temperatures $\{\beta_i\}_{i=1}^n$ does not lead anymore to a significant improvement in the corresponding sum of squares $\sum_{i=1}^{n-1}(\beta_i - \beta_{i+1})\ [g(\beta_{i+1}) - g(\beta_i)]$ which determines the acceptance rate in tempered transitions.

The second simulated annealing version searches over the constrained search space

$$\left\{\{\beta_i\}_{i=1}^n \in \{\beta_0\} \times [\beta_{\min}, \beta_0]^{n-2} \times \{\beta_{\min}\} : \quad \beta_i < \beta_{i+1} \quad \forall\, i = 1, \ldots, n-1\right\}$$

satisfying the ordering constraint $\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$ so that the initial inverse temperatures are chosen recursively by $\beta_i^{(0)} \sim U(\beta_{\min}, \beta_{i-1})$, $i = 2, \ldots, n-1$, where $\beta_1 = \beta_0$ and $\beta_n = \beta_{\min}$ are again fixed by definition. In analogy to the unconstrained algorithm, the component-wise update in the constrained version is defined by the following algorithm:

Algorithm 5.3:

Step 1 Draw $z \sim N(0, \sigma_T)$.

Step 2 Set

$$\beta_i^* = \begin{cases} \beta_i + z & \text{if } (\beta_i + z) \in [\beta_{i+1}, \beta_{i-1}], \\ 2\,\beta_{i+1} - (\beta_i + z) & \text{if } (\beta_i + z) < \beta_{i+1}, \\ 2\,\beta_{i-1} - (\beta_i + z) & \text{if } (\beta_i + z) > \beta_{i-1}. \end{cases}$$

Step 3 If $\beta_i^* \in [\beta_{i+1}, \beta_{i-1}]$, then set the proposal state $\beta_i' = \beta_i^*$. Otherwise go to Step 1.

Step 4 As this proposal is symmetric, accept β_i' with probability

$$\alpha(\beta_i, \beta_i') = \min\left\{1, \frac{\psi_T\left(\beta_i' \middle| \{\beta_k\}_{k \in \underline{n} \setminus \{i\}}\right)}{\psi_T\left(\beta_i \middle| \{\beta_k\}_{k \in \underline{n} \setminus \{i\}}\right)}\right\}.$$

Again the step size $\sigma_T = 10^{-3}$ works well in the later example (Section 6.3).

In general, simulated annealing is easy to implement because we can code it as a loop of MCMC steps in which, at every iteration, the temperature and the step size are adjusted according to the annealing schedule. The only problem is that we have to check the convergence of the algorithm, for example by comparing the results from multiple runs started at points from all over the search space. As the cost of simulated annealing is proportional to the length of each run times the number of runs, the optimisation comes at quite a high cost, which will be higher, the longer the time of convergence. We can avoid the convergence problem by carrying out an exhaustive search over a discretised version of the search space. We will present an efficient search algorithm in the next section.

5.4.3 Dynamic programming

Dynamic programming offers an efficient way to search for the optimal inverse temperatures $\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$ minimising

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1})\ [g(\beta_{i+1}) - g(\beta_i)]$$

provided that the search space is finite. We can define a finite search space by assuming that the n temperatures $\{\beta_i\}_{i=1}^n$ can only be placed on m available

positions $\{b_k\}_{k=1}^m$ satisfying $\beta_{\min} = b_m < \cdots < b_1 = \beta_0$ and $m > n$ so that the temperatures β_n and β_1 have already their fixed positions $\beta_n = b_m$ and $\beta_1 = b_1$. The ordering constraints imply that there are positions which a particular temperature can never take. For example, we cannot place β_3 on the position b_2 because if we did, we could not satisfy the constraint $\beta_3 < \beta_2 < \beta_1$ for there would be no position between $\beta_3 = b_2$ and $\beta_1 = b_1$ that β_2 could take. By similar considerations, we can deduce that each temperature β_i can only occupy the positions $b_{m-(n-i)}, \ldots, b_i$, $i = 2, \ldots, n-1$. We can use this information to set up the dynamic programming algorithm. Dynamic programming works recursively by going through the stages $j = 3, \ldots, n$ in ascending order. At the end of the $(j-1)$th stage, we have determined the optimal sets $\{\beta_{j-2}, \ldots, \beta_1\}$ and the corresponding minimal costs

$$C(\beta_{j-1}) = \sum_{i=1}^{j-2} (\beta_i - \beta_{i+1}) \, [g(\beta_{i+1}) - g(\beta_i)]$$

for *every* position $\beta_{j-1} \in \left\{b_{m-(n-(j-1))}, \ldots, b_{j-1}\right\}$. As the sets and the costs usually vary with the value of β_{j-1}, we have to store all this information if we want to tackle the jth stage. In the jth stage, we let β_j take all the possible values $b_{m-(n-j)}, \ldots, b_j$ one after another. If β_j currently occupies b_k, then we keep this position fixed until we have evaluated the cost

$$\begin{aligned} C(\beta_j) &= \sum_{i=1}^{j-1} (\beta_i - \beta_{i+1}) \, [g(\beta_{i+1}) - g(\beta_i)] \\ &= (\beta_{j-1} - \beta_j) \, [g(\beta_j) - g(\beta_{j-1})] + C(\beta_{j-1}) \end{aligned}$$

for *every* possible value $\beta_{j-1} = b_{k-1}, \ldots, b_{j-1}$ that β_{j-1} can take (following the ordering constraint $\beta_j < \beta_{j-1} < \cdots < \beta_1$). By trying all the possible β_{j-1} values, we will find the value that minimises $C(\beta_j)$ when $\beta_j = b_k$. As preparation for the $(j+1)$th stage, we will store the minimal cost $C(\beta_j)$ and the corresponding optimal temperatures $\{\beta_{j-1}, \ldots, \beta_1\}$ for $\beta_j = b_k$. Then we will move on to the next value $\beta_j = b_{k+1}$ and repeat the optimisation. At the end of stage j, we know the lowest cost and the underlying best sequence for every $\beta_j \in \left\{b_{m-(n-j)}, \ldots, b_j\right\}$. In the final stage, the nth stage, there is only one possible position that β_n can take, namely $\beta_n = b_n$. For this position, we can find the smallest cost $C(\beta_n)$ and the best temperatures $\{\beta_{n-1}, \ldots, \beta_1\}$ as before. By definition of the cost function, the minimal cost $C(\beta_n)$ is the desired minimal sum of squares $\sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \, [g(\beta_{i+1}) - g(\beta_i)]$ so that the last set of optimal temperatures is the set that we want to use in the tempered transitions algorithm.

Let us illustrate the principle of dynamic programming on the fictitious example of finding the optimal inverse temperatures $\beta_{\min} = \beta_4^* < \beta_3^* < \beta_2^* < \beta_1^*$ on six available positions $\beta_{\min} = b_6 < b_5 < \cdots < b_1$. Due to the end constraints, $\beta_1 = b_1$ and $\beta_4 = b_6$ are always fixed so that this optimisation problem has two free variables β_3 and β_2 where $\beta_3 \in \{b_5, b_4, b_3\}$ and $\beta_2 \in \{b_4, b_3, b_2\}$ subject to $\beta_3 < \beta_2$. In the first stage, all possible places of β_2 given β_3 are tried. First $\beta_3 = b_3$ is chosen. As $\beta_2 = b_2$ is the only possible position β_2 can take, the optimal inverse temperatures are $\beta_3^* = b_3$ and $\beta_2^* = b_2$:

$$\beta_3 = b_3 \qquad \Rightarrow \qquad \_\ \_\ \_\ \underline{\beta_3^*}\ \underline{\beta_2^*}\ \underline{\beta_1^*} \quad \text{(I)}.$$

Then $\beta_3 = b_2$ is tried so that either $\beta_2 = b_3$ or $\beta_2 = b_2$ is optimal. We will assume that the optimal solution is $\beta_3^* = b_4$ and $\beta_2^* = b_3$:

$$\beta_2 = b_3 \qquad \Rightarrow \qquad \begin{cases} \_\ \_\ \underline{\beta_3}\ \_\ \underline{\beta_2}\ \underline{\beta_1} & \text{(IIa)} \\ \_\ \_\ \underline{\beta_3^*}\ \underline{\beta_2^*}\ \_\ \underline{\beta_1^*} & \text{(IIb)}. \end{cases}$$

Finally $\beta_3 = b_5$ is chosen so that $\beta_2 = b_4, b_3, b_2$ are possible. Let us assume that the best choice is $\beta_3^* = b_4$ and $\beta_2^* = b_3$:

$$\beta_3 = b_5 \qquad \Rightarrow \qquad \begin{cases} \_\ \underline{\beta_3}\ \_\ \_\ \underline{\beta_2}\ \underline{\beta_1} & \text{(IIIa)} \\ \_\ \underline{\beta_3^*}\ \_\ \underline{\beta_2^*}\ \_\ \underline{\beta_1^*} & \text{(IIIb)} \\ \_\ \underline{\beta_3}\ \underline{\beta_2}\ \_\ \_\ \underline{\beta_1} & \text{(IIIc)}. \end{cases}$$

In the second stage, we find the optimal temperatures $(\beta_3, \beta_2, \beta_1)$ for every position that β_4 can occupy. As β_4 is the last temperature, it can only take the position $\beta_4 = b_6$. To find the best sequence $(\beta_4, \beta_3, \beta_2, \beta_1)$, we only have to vary β_3 and place β_2 on its optimal position obtained in the previous stage. This narrows the search down to the following options:

$$\beta_4 = b_6 \qquad \Rightarrow \qquad \begin{cases} \underline{\beta_4}\ \_\ \_\ \underline{\beta_3}\ \underline{\beta_2}\ \underline{\beta_1} & \text{using optimal (I)} \\ \underline{\beta_4}\ \_\ \underline{\beta_3}\ \underline{\beta_2}\ \_\ \underline{\beta_1} & \text{using optimal (IIb)} \\ \underline{\beta_4^*}\ \underline{\beta_3^*}\ \_\ \underline{\beta_2^*}\ \_\ \underline{\beta_1^*} & \text{using optimal (IIIb)}. \end{cases}$$

Let us assume that the last option $\beta_4 = b_6$, $\beta_3 = b_5$, $\beta_2 = b_1$ and $\beta_1 = b_1$ is best. As the final stage is reached, this option is also the optimal scheme for tempered transitions.

Implementing dynamic programming is not difficult. One advantage is that it is a very efficient exhaustive search method. Another is that it always finds the global optimum. Dynamic programming may however require a lot of storage since it needs to remember the $(j-1)$ optimal inverse temperatures

(and the associated cost) for every of the $(m-n+1)$ positions which the jth inverse temperature can take so that $(n-2)\times(m-n)$ values need to be stored in preparation for the final stage. Since dynamic programming is a deterministic algorithm, we can calculate its total cost. Recall that, at the jth stage, $j=3,\ldots,(n-1)$, β_j is tried on all the $(m-n+1)$ possible positions $b_{m-(n-j)},\ldots,b_j$. If β_j currently occupies b_k, then β_{j-1} is tried on all the $(k-j+1)$ possible positions $\beta_{j-1}=b_{k-1},\ldots,b_{j-1}$ so that the cost at the jth stage, $j=3,\ldots,(n-1)$, is proportional to

$$\sum_{k=j}^{m-(n-j)}(k-j+1)=\sum_{l=1}^{m-n+1}l$$
$$=\frac{1}{2}(m-n+1)(m-n+2).$$

This cost has to be multiplied by $(n-3)$ because it occurs at the $(n-3)$ stages $j=3,\ldots,(n-1)$. Also, the cost of the final stage has to be added: at the nth stage, $\beta_n=b_m$ is the only possibility so that β_{n-1} is the only parameter that is moved. As β_{n-1} can occupy the $(m-n+1)$ positions $\beta_{n-1}=b_{m-1},\ldots,b_{n-1}$, the cost of the last stage is proportional to $(m-n+1)$. Adding all these costs together yields a total cost proportional to

$$\frac{1}{2}(n-3)(m-n+1)(m-n+2)+(m-n+1).$$

That means that the total cost is of order $\mathcal{O}(m^2)$ when n is fixed and of order $\mathcal{O}(n^3)$ when m is fixed.

5.5 Summary

In this chapter, we have discussed that the aim of tuning temperatures in tempered transitions is increasing the efficiency of the algorithm. If we use more temperatures, then we will also raise the cost so that we should only set more temperatures if the gain in mixing is worth the additional cost. If we fix the number of temperatures as well as the hottest temperature, then the true optimisation problem is to maximise the expected acceptance probability $\mathbb{E}_\varphi(\alpha)$. We can seldom tackle the true problem directly, but we can approach it indirectly by solving the related problem of minimising the sum of squares $\sum_{i=0}^{n-1}(\beta_i-\beta_{i+1})\,[g(\beta_{i+1})-g(\beta_i)]$ which is based on the "ideal world" assumption that the Markov transition kernels $\{T_{\beta_i}\}_{i=1}^n$ of the secondary chain generate independent samples from the corresponding distributions $\{p_{\beta_i}\}_{i=1}^n$. To simplify the search for the optimal temperatures, we have proven that these temperatures satisfy the ordering constraint $\beta_{\min}=\beta_n<\cdots<\beta_1=\beta_0$.

We were able to optimise the temperatures analytically in the toy example of sampling from a multivariate normal distribution. In this example, the default choice of geometrically spaced temperatures proved to be optimal. As the analytic optimisation is in general not possible, we suggested two alternative optimisation methods, namely simulated annealing and dynamic programming. In the next chapter, we will test the tuning technique in a rare toy example in which the true optimisation problem can actually be tackled under the "ideal world" assumption. This will give us the opportunity to investigate how well we approach the true problem by the related problem and what happens if the "ideal world" assumption is not met.

Chapter 6

Testing the Tuning Technique on a Toy Example

6.1 Introduction

In the following, we will use the tuning technique developed in the previous chapter to design an efficient tempered transitions method for the sampler-unfriendly simplified Witch's Hat distribution. First, we will introduce the distribution (Section 6.2). Then, we will check how suitable simulated annealing and dynamic programming are for finding the temperature sequence $\{\beta_i\}_{i=1}^n$ that minimises the sum of squares $\sum_{i=1}^{n-1}(\beta_i - \beta_{i+1})\,[g(\beta_{i+1}) - g(\beta_i)]$ where $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ (Section 6.3). After that, we will investigate how well our optimisation criterion holds up against the usually intractable goal of maximising the expected acceptance probability in comparison to the alternative of spacing temperatures geometrically (Section 6.4). Finally, we will assess the benefits of carrying out an optimisation based on idealising assumptions in cases in which these assumptions are not met (Section 6.5) before closing the chapter with a summary (Section 6.6).

6.2 Simplified Witch's Hat

The original Witch's Hat distribution was introduced by Matthews (1993) as a cautionary example for Gibbs sampling. Its parameters can be chosen such that a Gibbs sampler fails to converge within any reasonable amount of time. Due to the shape of the distribution, the lack of convergence is very hard to detect. The Witch's Hat is defined on the d-dimensional open unit cube $C = (0, 1)^d$ as the mixture of a uniform and a normal distribution with the

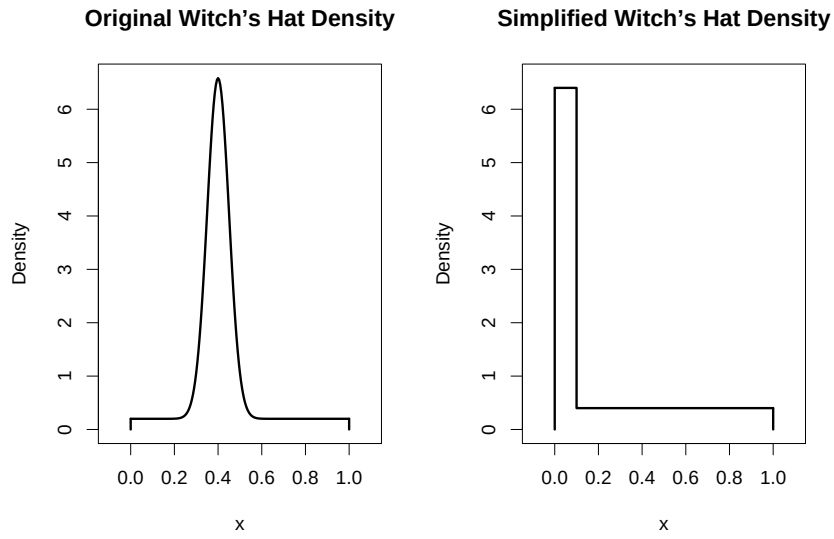

Figure 6-1: The original Witch's Hat distribution (here with parameters $\delta = 0.2$, $y = 0.4$, $\sigma = 0.05$) looks indeed like a witch's hat *(left)*, while the simplified Witch's Hat distribution (here with parameters $a = 0.1$, $b = 15$) looks more like an "L" than a witch's hat *(right)*.

normal component being the "peak" and the uniform component being the "brim" of a witch's hat (see Figure 6-1). For fixed $\delta \in (0, 1)$ and $\sigma > 0$, the Witch's Hat distribution is given by

$$p(\theta|y) \propto (1-\delta)\ \left(2\pi\sigma^2\right)^{-d/2} \exp\left[-\frac{1}{2\,\sigma^2}\sum_{i=1}^{d}(y_i-\theta_i)^2\right] + \delta\,\mathbb{1}_{\{y\in C\}} \qquad \text{for } \theta \in C$$

where y is a single multivariate observation in C. Although the Witch's Hat distribution is not multimodal, it is notorious for the mixing problems it causes if the peak contains a lot of probability mass but is very hard to hit by standard MCMC. In this case, the mixing problem can go either way: either the sampler is trapped in the peak and cannot visit the brim although this part has significant probability or the sampler moves around the brim and cannot detect the peak because the latter is concentrated on a small spot. Due to these sampling difficulties, the Witch's Hat is in general a good test problem for MCMC methods. The Witch's Hat distribution can be simplified to a mixture of a uniform on $[0, 1]^d$ (brim) and a uniform on $[0, a]^d$ (peak) without losing its intrinsic sampling difficulty (Geyer and Thompson 1995). For the following work, it is sufficient to consider the one-dimensional simplified Witch's Hat

with parameters $0 < a < 1$ and $0 < b$:

$$\begin{aligned} p_\beta(x) &= \frac{1}{a\,(1+b)^\beta + (1-a)} \left(1 + b\,\mathbb{1}_{\{x \le a\}}\right)^\beta \\ &= \frac{1}{a\,(1+b)^\beta + (1-a)} \exp\left\{ -\beta \underbrace{\left[-\log\left(1 + b\,\mathbb{1}_{\{x \le a\}}\right)\right]}_{=:\,h(x)} \right\}, \qquad \text{for } 0 \le x \le 1. \end{aligned}$$

The one-dimensional version actually resembles more the capital letter "L" than a witch's hat (see Figure 6-1). The curve $g(\beta) = \mathbb{E}_{p_\beta}\left[h(X)\right]$ is then

$$\begin{aligned} g(\beta) &= \frac{1}{a\,(1+b)^\beta + (1-a)} \left[\int_0^a (-1)\,(1+b)^\beta\,\log(1+b)\,\mathrm{d}x + \int_a^1 (-1)\,1^\beta \underbrace{\log(1)}_{=0}\,\mathrm{d}x \right] \\ &= \frac{(-a)\,(1+b)^\beta \log(1+b)}{a\,(1+b)^\beta + (1-a)}. \end{aligned} \tag{6.1}$$

The Witch's Hat is a good example to demonstrate that the curve $g(\beta)$ can be convex, concave or a mixture of both as mentioned in Section 5.3.1. As the shape depends on the second derivative of $g(\beta)$, we will derive the first two derivatives. Its first derivative is

$$\begin{aligned} g'(\beta) &= \frac{\left[a\,(1+b)^\beta + (1-a)\right](-a)\,(1+b)^\beta\,[\log(1+b)]^2 + \left[a\,(1+b)^\beta \log(1+b)\right]^2}{\left[a\,(1+b)^\beta + (1-a)\right]^2} \\ &= \frac{a\,(1+b)^\beta\,[\log(1+b)]^2\left[-a\,(1+b)^\beta - (1-a) + a\,(1+b)^\beta\right]}{\left[a\,(1+b)^\beta + (1-a)\right]^2} \\ &= \frac{a\,(a-1)\,(1+b)^\beta\,[\log(1+b)]^2}{\left[a\,(1+b)^\beta + (1-a)\right]^2}. \end{aligned}$$

To calculate the second derivative, we need

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{(1+b)^\beta}{\left[a\,(1+b)^\beta + (1-a)\right]^2}\right] \\ &= \frac{\left[a\,(1+b)^\beta + (1-a)\right](1+b)^\beta \log(1+b) - 2\,(1+b)^\beta\left[a\,(1+b)^\beta \log(1+b)\right]}{\left[a\,(1+b)^\beta + (1-a)\right]^3} \\ &= \frac{(1+b)^\beta \log(1+b)\left[a\,(1+b)^\beta + (1-a) - 2\,a\,(1+b)^\beta\right]}{\left[a\,(1+b)^\beta + (1-a)\right]^3} \\ &= \frac{(-1)\,(1+b)^\beta \log(1+b)\left[a\,(1+b)^\beta - (1-a)\right]}{\left[a\,(1+b)^\beta + (1-a)\right]^3}. \end{aligned}$$

Note that the second line does not include one of the factors $\left[a\,(1+b)^{\beta}+(1-a)\right]$ due to cancellation. The second derivative then follows:

$$\begin{aligned} g''(\beta) &= a\,(a-1)\,[\log(1+b)]^2\,\frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{(1+b)^{\beta}}{\left[a\,(1+b)^{\beta}+(1-a)\right]^2}\right] \\ &= \underbrace{\frac{(-a)\,(a-1)\,(1+b)^{\beta}\,[\log(1+b)]^3}{\left[a\,(1+b)^{\beta}+(1-a)\right]^3}}_{>0}\,\left[a\,(1+b)^{\beta}-(1-a)\right]. \end{aligned}$$

In the last expression, the fraction is positive because both the numerator and the denominator are positive (for $a \in (0,1)$ and $b > 0$). In consequence, the shape of the curve $g(\beta)$ is determined by the last factor $\left[a\,(1+b)^{\beta}-(1-a)\right]$ of the second derivative $g''(\beta)$. If this factor is positive, the second derivative is also positive so that the curve is convex. Similarly, if this factor is negative, the curve is concave.

Let us now discuss for which parameters a and b the curve $g(\beta)$ is convex, concave or a mixture of both. First we will prove that the parameter choice $a \in \left(\frac{1}{2}, 1\right)$ and $0 < b$ leads to a positive factor $\left[a\,(1+b)^{\beta}-(1-a)\right] > 0$ which implies that the second derivative $g''(\beta) > 0$ is positive so that the curve $g(\beta)$ is convex on $[0,1]$:

$$\begin{array}{rrcll} & a & \in & \left(\frac{1}{2},1\right) & \text{(and } b>0\text{)} \\ \Leftrightarrow & 1 & > & \frac{1-a}{a} & \\ \Leftrightarrow & 0 & > & \log\left(\frac{1-a}{a}\right) & \\ \Leftrightarrow & \beta\log(1+b) & > & \log\left(\frac{1-a}{a}\right) & \forall\ \beta\in[0,1] \quad \text{(as } b>0\text{)} \\ \Leftrightarrow & a\,(1+b)^{\beta}-(1-a) & > & 0 & \forall\ \beta\in[0,1] \\ \Leftrightarrow & g''(\beta) & > & 0 & \forall\ \beta\in[0,1]. \end{array}$$

Similarly, we obtain a concave curve if $a \in \left(0, \frac{1}{2}\right)$ and $b \in \left(0, \frac{1-2a}{a}\right)$ as then $\left[a\,(1+b)^{\beta}-(1-a)\right] < 0$ and thus $g''(\beta) < 0$ for all $\beta \in [0,1]$:

$$\begin{array}{rrcll} & b & < & \frac{1-2a}{a} & \left[\text{where } \frac{1-2a}{a} > 0 \text{ as } a \in \left(0,\frac{1}{2}\right)\right] \\ \Leftrightarrow & 1+b & < & \frac{1-a}{a} & \\ \Leftrightarrow & \beta\log(1+b) & < & \log\left(\frac{1-a}{a}\right) & \forall\ \beta\in[0,1] \\ \Leftrightarrow & a\,(1+b)^{\beta}-(1-a) & < & 0 & \forall\ \beta\in[0,1] \\ \Leftrightarrow & g''(\beta) & < & 0 & \forall\ \beta\in[0,1]. \end{array}$$

Otherwise, the curve will show both convex and concave behaviour on $[0,1]$. Examples for each behaviour and the underlying Witch's Hat densities are plotted in Figures 6-2 and 6-3. Having introduced the toy example, we can now investigate how the optimisation methods perform in the next section.

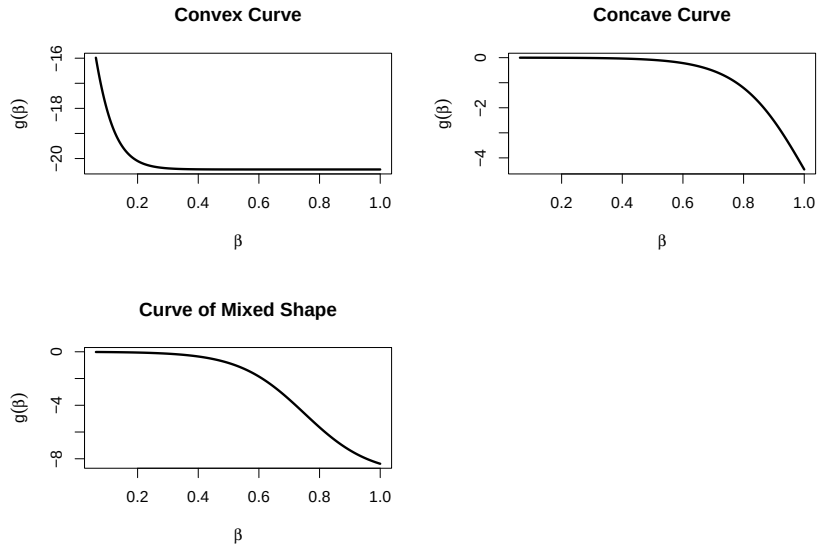

Figure 6-2: *Top left:* An example of a convex curve $g(\beta)$, here induced by the simplified Witch's Hat with parameters $a = 0.5$ and $b = 7.5 \cdot 10^8$. *Top right:* An example of a concave curve $g(\beta)$, here induced by the simplified Witch's Hat with parameters $a = 10^{-4}$ and $b = 9.5 \cdot 10^3$. *Bottom left:* An example of a curve $g(\beta)$ of mixed shape, here induced by the simplified Witch's Hat with parameters $a = 10^{-3}$ and $b = 10^4$.

6.3 Testing simulated annealing and dynamic programming

We will test the optimisation methods simulated annealing and dynamic programming when the Witch's parameters are set to $a = 10^{-3}$ and $b = 10^4$. We want to find the best temperature schedule $\beta_{\min} = \beta_n < \ldots < \beta_1 = \beta_0$ yielding the smallest sum of squares

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i)\right]$$

where $g(\beta)$ is given by (6.1). We will use $\beta_{\min} = \frac{1}{16}$ as the hottest temperature because it encourages the sampler to jump from the peak into the brim. For comparison, in a Metropolis update, the acceptance probability of moving from $x \leq a$ (peak) to $x' > a$ (brim)

$$\begin{aligned} \alpha(x, x') &= \min\left\{1, \frac{p_\beta(x')}{p_\beta(x)}\right\} \\ &= \min\left\{1, \left(1 + b\, \mathbb{1}_{\{x \leq a\}}\right)^{-\beta}\right\} \end{aligned}$$

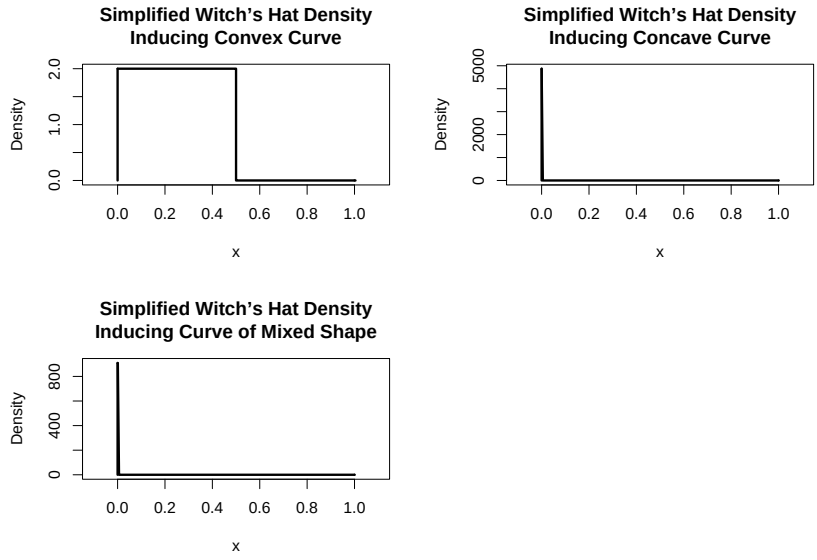

Figure 6-3: *Top left:* An example of a simplified Witch's Hat inducing a convex curve $g(\beta)$, here with parameters $a = 0.5$ and $b = 7.5 \cdot 10^8$. *Top right:* An example of a simplified Witch's Hat inducing a concave curve $g(\beta)$, here with parameters $a = 10^{-4}$ and $b = 9.5 \cdot 10^3$. *Bottom left:* An example of a simplified Witch's Hat inducing a curve $g(\beta)$ of mixed shape, here with parameters $a = 10^{-3}$ and $b = 10^4$.

is $\alpha(x, x') = \frac{1}{10\,001}$ at $\beta_0 = 1$ and $\alpha(x, x') = \left(\frac{1}{10\,001}\right)^{1/16} = 0.562$ at $\beta_{\min} = \frac{1}{16}$ in this particular example. This implies that standard MCMC needs on average 10 000 attempts to leave the peak under β_0 and two attempts under $\beta_{\min}$. In addition, the mass under the peak

$$\begin{aligned}\mathbb{P}_{p_\beta}\{X \leq a\} &= \int_0^a \frac{(1+b)^\beta}{a(1+b)^\beta + (1-a)}\,\mathrm{d}x \\ &= \frac{a\,(1+b)^\beta}{a\,(1+b)^\beta + (1-a)}\end{aligned}$$

reduces from 0.909 (at β_0) to 0.002 (at $\beta_{\min}$) so that the sampler is free to explore the brim, which also improves the mixing.

The optimisation methods are set up as described in Sections 5.4.2 and 5.4.3. That means that both versions of simulated annealing (unconstrained/constrained search space) are run based on a logarithmic annealing schedule $\{T_k\}_{k=1}^N$ between $T_1 = \frac{1}{2}$ and $T_N = 10^{-306}$ with constant step size $\sigma_T = 10^{-3}$ at each annealing temperature T. We vary $N = 10\,000, 100\,000$ to see the effect of the schedule length. Similarly, in dynamic programming, we search over $m = 1\,001, 5\,001, 10\,001$ equidistant positions between $\beta_{\min}$ and β_0

ordering constraint	iterations	20 Inverse Temperatures $S(\{\beta_i^{\text{approx}}\})$	δ_β	time in sec
no	100 000	0.3163133	4.4×10^{-6}	139
	10 000	0.3163140	4.1×10^{-4}	14
yes	100 000	0.3163133	2.2×10^{-6}	29
	10 000	0.3163140	4.1×10^{-4}	3

Table 6-1: Simulated annealing was run to optimise the sum of squares S for $n = 20$ inverse temperatures. The variation in accuracy between methods is insignificant. The time is taken for a single run. Usually, replicate runs are required for convergence tests in which case the noted time has to be multiplied by the number of replicate runs.

ordering constraint	#possible positions	20 Inverse Temperatures $S(\{\beta_i^{\text{approx}}\})$	δ_β	time in sec
yes	10 001	0.3163135	1.1×10^{-4}	150
	5 001	0.3163140	1.1×10^{-4}	36
	1 001	0.3163304	8.8×10^{-4}	1

Table 6-2: Dynamic programming was run to optimise the sum of squares S for $n = 20$ inverse temperatures. The variation in accuracy between methods is insignificant.

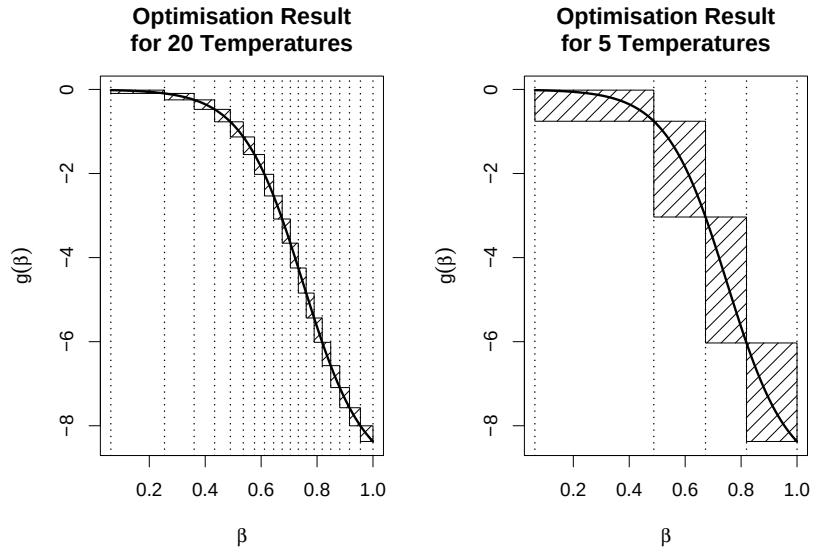

Figure 6-4: The figure shows the minimal sum of squares for $n = 20$ *(left)* and $n = 5$ *(right)* temperatures between $\beta_n = \frac{1}{16}$ and $\beta_1 = 1$ when the simplified Witch's Hat takes parameters $a = 0.5$ and $b = 7.5 \cdot 10^8$.

to change the accuracy of the results. We will compare the performance of all these methods in two cases, when optimising $n = 5$ and $n = 20$ temperatures. In each case, we need a benchmark set $\{\beta_i^*\}_{i=1}^n$. To find one, ten replicate simulated annealing runs (unconstrained search, $N = 100\,000$) are carried out. As all runs yield the same sum of squares (namely $S = 0.3163133$ for $n = 20$ and $S = 1.598644$ for $n = 5$), but differ slightly in the corresponding temperature values, the benchmark set is defined by the pooled mean of the replicates. Let $\beta_i^{(r)}$ denote the ith inverse temperatures returned by the rth replicate run, then the estimated ith true inverse temperature, $i = 2, \ldots, n-1$, is set to be

$$\beta_i^* = \frac{1}{R} \sum_{r=1}^{R} \beta_i^{(r)}$$

where R is the total number of replicate runs, and where $\beta_1^* = 1$ and $\beta_n^* = \beta_{\min}$ by definition. To assess the spread of the results, the relative error

$$\delta_\beta = \left(\frac{\sum_{i=0}^{n} \left(\beta_i^* - \beta_i^{\text{approx}}\right)^2}{\sum_{i=0}^{n} \left(\beta_i^*\right)^2} \right)^{\frac{1}{2}} \tag{6.2}$$

is calculated with respect to each replicate set $\{\beta_i^{\text{approx}}\}_{i=1}^n$. The deviation is tiny; the largest value is $2.9 \cdot 10^{-6}$ when $n = 20$ and $6.9 \cdot 10^{-8}$ when $n = 5$. We can also use δ_β to assess the difference between the benchmark set $\{\beta_i^*\}$ and any other set $\{\beta_i^{\text{approx}}\}$ of interest, for example the optimal

set returned by the various optimisation methods. The optimisation results are displayed in Tables 6-1 to 6-4. The benchmark solutions for $n = 5$ and $n = 20$ temperatures are plotted in Figure 6-4. In general, we can say that, as expected, simulated annealing gains accuracy as the length N increases, while dynamic programming improves as the mesh size decreases. Although the accuracy varies between all the methods, this variation does not matter because all the errors are negligible in both δ_β and sum of squares. It is worth using the constrained version of simulated annealing when $n = 20$ because it reduces the computational cost in this example significantly. When $n = 5$, the costs of the unconstrained and the constrained search are similar because the spacing between temperatures (see Figure 6-4) is much larger than the step size $\sigma_T = 10^{-3}$ so that the constraint that every proposal should lie between its adjacent neighbours is met even if the constraint is not explicitly implemented. Another point is that dynamic programming seems to be slightly more expensive than simulated annealing. However, if we bear in mind that we usually have to run replicate runs to diagnose the convergence of simulated annealing, in which case the noted time has to be multiplied by the number of replicate runs, then dynamic programming becomes the less expensive method for this example. Overall, simulated annealing and dynamic programming perform well when optimising the sum of squares. Since dynamic programming is easier to handle and since it does not require any convergence tests, it will be used as the preferred method in the following work.

6.4 How closely does the related optimisation problem approximate the true one?

In this section, we will investigate whether optimising the sum of squares

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i)\right]$$

brings us closer to the true goal of maximising the expected acceptance probability

$$\mathbb{E}_\varphi \left[\alpha(X_0, \ldots, X_{n-1}, X_n, X'_{n-1}, \ldots, X'_0)\right]$$

than Neal's suggested geometric default. We will conduct the comparison under the "ideal world" assumption that the auxiliary states $X_1, \ldots, X_n$ and $X'_{n-1}, \ldots, X'_0$ are independent samples from the equilibrium distribution under

		5 Inverse Temperatures		
ordering constraint	iterations	$S(\{\beta_i^{\text{approx}}\})$	δ_β	time in sec
no	100 000	1.598644	2.4×10^{-8}	6
	10 000	1.598644	4.6×10^{-7}	1
yes	100 000	1.598644	2.4×10^{-8}	6
	10 000	1.598644	1.2×10^{-6}	1

Table 6-3: Simulated annealing was run to optimise the sum of squares S for $n = 5$ inverse temperatures. The variation in accuracy between methods is insignificant. The time is taken for a single run. Usually, replicate runs are required for convergence tests in which case the noted time has to be multiplied by the number of replicate runs.

		5 Inverse Temperatures		
ordering constraint	#possible positions	$S(\{\beta_i^{\text{approx}}\})$	δ_β	time in sec
yes	10 001	1.598644	2.4×10^{-6}	18
	5 001	1.598644	8.8×10^{-5}	4
	1 001	1.598650	2.7×10^{-4}	0

Table 6-4: Dynamic programming was run to optimise the sum of squares S for $n = 5$ inverse temperatures. The variation in accuracy between methods is insignificant.

which they are generated, in which case

$$\begin{aligned}\varphi(x_0,\ldots,x_n,x'_{n-1},\ldots,x'_0)\\ \propto p_{\beta_0}(x_0)\left[\prod_{i=1}^{n}p_{\beta_i}(x_i)\right]\left[\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x'_i)\right]\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}_\varphi(\alpha) &= \mathbb{E}_\varphi\left\{\min\left\{1,\left[\prod_{i=0}^{n-1}\frac{p_{\beta_{i+1}}(X_i)}{p_{\beta_i}(X_i)}\right]\left[\prod_{i=0}^{n-1}\frac{p_{\beta_i}(X'_i)}{p_{\beta_{i+1}}(X'_i)}\right]\right\}\right\}\\ &= \int_0^1\mu(\mathrm{d}x_0)\int_0^1\mu(\mathrm{d}x_1)\cdots\int_0^1\mu(\mathrm{d}x_{n-1})\int_0^1\mu(\mathrm{d}x'_{n-1})\cdots\int_0^1\mu(\mathrm{d}x'_1)\int_0^1\mu(\mathrm{d}x'_0)\\ &\quad\min\left\{1,\left[\prod_{i=0}^{n-1}\frac{p_{\beta_{i+1}}(x_i)}{p_{\beta_i}(x_i)}\right]\left[\prod_{i=0}^{n-1}\frac{p_{\beta_i}(x'_i)}{p_{\beta_{i+1}}(x'_i)}\right]\right\}\prod_{i=0}^{n-1}p_{\beta_i}(x_i)\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x'_i).\\ &= \int_0^1\mu(\mathrm{d}x_0)\int_0^1\mu(\mathrm{d}x_1)\cdots\int_0^1\mu(\mathrm{d}x_{n-1})\int_0^1\mu(\mathrm{d}x'_{n-1})\cdots\int_0^1\mu(\mathrm{d}x'_1)\int_0^1\mu(\mathrm{d}x'_0)\\ &\quad\min\left\{\prod_{i=0}^{n-1}p_{\beta_i}(x_i)\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x'_i),\ \prod_{i=0}^{n-1}p_{\beta_i}(x'_i)\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x_i)\right\}.\end{aligned}$$

Due to the simple form of the tempered Witch's Hat distributions

$$p_\beta(x)=\frac{1}{a\,(1+b)^\beta+(1-a)}\left(1+b\,\mathbb{1}_{\{x\le a\}}\right)^\beta,\qquad\text{for } 0\le x\le 1,$$

we can calculate the above integral analytically. Each of the distributions $p_{\beta_i}(x)$, $i=0,\ldots,n-1$, is piecewise constant, namely on $[0,a]$ and on $(a,1]$, so that the integrand

$$\min\left\{\prod_{i=0}^{n-1}p_{\beta_i}(x_i)\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x'_i),\ \prod_{i=0}^{n-1}p_{\beta_i}(x'_i)\prod_{i=0}^{n-1}p_{\beta_{i+1}}(x_i)\right\}$$

is also piecewise constant, namely on each product sub-space $\bigtimes_{k=1}^{2n}A_k$ with $A_k\in\{[0,a],(a,1]\}$, $k=1,\ldots,2n$. That means that we can calculate the integral by partitioning the product space into all the 2^{2n} possible sub-spaces $\bigtimes_{k=1}^{2n}A_k$ and integrate over each of them. In the simplest case ($n=1$), we compute the integral by

$$\begin{aligned}\int_0^1\int_0^1 f(x_0,x'_0)\mathrm{d}x_0\mathrm{d}x'_0 &= \int_0^a\int_0^a f(x_0,x'_0)\mathrm{d}x_0\mathrm{d}x'_0+\int_0^a\int_a^1 f(x_0,x'_0)\mathrm{d}x_0\mathrm{d}x'_0\\ &\quad+\int_a^1\int_0^a f(x_0,x'_0)\mathrm{d}x_0\mathrm{d}x'_0+\int_a^1\int_a^1 f(x_0,x'_0)\mathrm{d}x_0\mathrm{d}x'_0\end{aligned}$$

where $f(x_0,x'_0)=\min\left\{p_{\beta_0}(x_0)p_{\beta_1}(x'_0),\ p_{\beta_1}(x_0)p_{\beta_0}(x'_0)\right\}$. For greater n values, the integration is tedious so that it is best to set up a computer program which can find a suitable partition and integrate over it. Since we can compute $\mathbb{E}_\varphi(\alpha)$ for various choices of $\{\beta_i\}$, we can find the best temperature schedule by an exhaustive search over a discretised search space subject to the optimality

constraint $\beta_{\min} = \beta_n < \ldots < \beta_1 = \beta_0$. Using the same discretised search space, we can obtain the optimal sequence minimising the sum of squares by dynamic programming as before. We can assess how well the solution to the related problem (or alternatively the geometric solution) approximates the truth by the relative accuracies $(1 - \delta_\beta)$ and $\left(1 - \delta_{\mathbb{E}(\alpha)}\right)$ where

$$\delta_\beta = \left(\frac{\sum_{i=0}^{n} \left(\beta_i^{\text{true}} - \beta_i^{\text{approx}}\right)^2}{\sum_{i=0}^{n} \left(\beta_i^{\text{true}}\right)^2} \right)^{\frac{1}{2}} \qquad \text{and} \qquad \delta_{\mathbb{E}(\alpha)} = \frac{\mathbb{E}_{\text{true}}(\alpha) - \mathbb{E}_{\text{approx}}(\alpha)}{\mathbb{E}_{\text{true}}(\alpha)}.$$

In the above terms, $\{\beta_i^{\text{true}}\}$ and $\mathbb{E}_{\text{true}}(\alpha)$ denote the true optimal solutions, while $\{\beta_i^{\text{approx}}\}$ and $\mathbb{E}_{\text{approx}}(\alpha)$ are their approximations.

We are interested in seeing how much closer our tuning technique comes to the true solution than the alternative geometric spacing. If it is true that the sum of squares and the acceptance rate are somehow related, then we would expect that our criterion does better, the more the shape of the curve differs from the shape of the curve $g(\beta) = \frac{1}{2\beta}$ for which we have already shown that the geometric spacing is optimal in Section 5.4.1 (see also Figure 5-6 for a picture of that curve). Hence, we can check our intuition by carrying out the closeness test in examples of the three different scenarios (convex curve, concave curve and curve of mixed behaviour). The results are summarised in Tables 6-5 to 6-7. As expected, our tuning technique does always better than the geometric spacing. Its advantage is greater, the greater the difference in shape between the underlying curve and $g(\beta) = \frac{1}{2\beta}$. The geometric solution does quite well in the convex example, but falls significantly behind the tuning technique in the other two examples. This behaviour is illustrated for the three cases in the Figures 6-5 to 6-7 where the true optimal temperature sequence, the related scheme and the geometric schedule and their corresponding sum of squares are plotted. From these figures, we can also learn how to space the temperatures if we want to obtain a small sum of squares: the spacing between inverse temperatures should be smaller, the greater the slope of the curve so that there are more temperatures in areas of strong decay than in areas of weak decay. We can use this knowledge to predict by inspection whether a given temperature scheme, such as the geometric, will be a sensible choice. It is worth taking a closer look at the tables. The tuning technique approximates the optimiser $\{\beta_i^{\text{true}}\}$ and the maximum $\mathbb{E}_{\text{true}}(\alpha)$ quite well in all examples. The geometric scheme has in comparison a relatively poor accuracy with respect to $\{\beta_i^{\text{true}}\}$, but achieves nevertheless a relatively good acceptance rate in the convex case $\left[\left(1 - \delta_{\mathbb{E}(\alpha)}\right) = 0.98\right]$ and in the concave case $\left[\left(1 - \delta_{\mathbb{E}(\alpha)}\right) = 0.8\right]$. But this is not

Closeness for Convex Curve

temperature scheme	$\mathbb{E}_\varphi(\alpha)$	$\left(1-\delta_{\mathbb{E}(\alpha)}\right)$	$(1-\delta_\beta)$
geometric	0.79	0.98	0.70
optimal	0.80	1.00	0.91
true	0.81	–	–

Table 6-5: Closeness of the approximations to the true solution in an example where the curve $g(\beta)$ is convex ($a = 0.5$ and $b = 7.5 \cdot 10^8$). "Geometric" refers to the geometric default scheme, "optimal" to the sequence with the smallest sum of squares and "true" to the scheme maximising $\mathbb{E}_\varphi(\alpha)$.

always the case as the mixed example shows. In this example, the geometric scheme yields a poor relative accuracy of $\left(1-\delta_{\mathbb{E}(\alpha)}\right) = 0.39$. Intuitively, one may have thought that the relative accuracy should have been better than that in the concave curve because the concave shape is the worst fit to the model shape of $g(\beta) = \frac{1}{2\beta}$. This is therefore a good example to demonstrate that we cannot predict the accuracy from the sum of squares because there is no simple relation between the sum of squares and the acceptance rate as discussed in Section 5.2.3. The mixed example (where $a = 10^{-3}$ and $b = 10^4$) shows that a lot of efficiency can be gained by optimisation because the acceptance rate can be more than doubled.

In summary, we have seen that the tuning technique indeed approximates the true solution and that the Witch's Hat with parameters $a = 10^{-3}$ and $b = 10^4$ is a counter-example to the assumption that geometric inverse temperatures always perform satisfactorily. We will use the counter-example to investigate whether the optimal scheme can hold its advantage over the geometric scheme in the "real world" of slow convergence at each temperature.

6.5 Benefit of optimisation in the real world scenario

In this section, we will test whether the optimisation which is carried out under the ideal world assumption of instant convergence at each temperature is of any benefit in the real world scenario of slow convergence. To assess the benefit, we will monitor the acceptance rate and the integrated autocorrelation time

Closeness for Concave Curve

temperature scheme	$\mathbb{E}_\varphi(\alpha)$	$\left(1-\delta_{\mathbb{E}(\alpha)}\right)$	$(1-\delta_\beta)$
geometric	0.52	0.80	0.47
optimal	0.62	0.96	0.91
true	0.65	–	–

Table 6-6: Closeness of the approximations to the true solution in an example where the curve $g(\beta)$ is concave ($a = 10^{-4}$ and $b = 9.5 \cdot 10^3$). "Geometric" refers to the geometric default scheme, "optimal" to the sequence with the smallest sum of squares and "true" to the scheme maximising $\mathbb{E}_\varphi(\alpha)$.

Closeness for Curve of Mixed Shape

temperature scheme	$\mathbb{E}_\varphi(\alpha)$	$\left(1-\delta_{\mathbb{E}(\alpha)}\right)$	$(1-\delta_\beta)$
geometric	0.17	0.39	0.54
optimal	0.41	0.95	0.93
true	0.43	–	–

Table 6-7: Closeness of the approximations to the true solution in an example where the curve $g(\beta)$ is of mixed shape ($a = 10^{-4}$ and $b = 10^4$). "Geometric" refers to the geometric default scheme, "optimal" to the sequence with the smallest sum of squares and "true" to the scheme maximising $\mathbb{E}_\varphi(\alpha)$.

$\tau(x)$ estimated by (2.3) with respect to the theoretical mean

$$\begin{aligned}\mathbb{E}_{p_{\beta_0}}(X) &= \int_0^a \frac{x\,(1+b)}{a(1+b)+(1-a)}\,\mathrm{d}x + \int_a^1 \frac{x}{a(1+b)+(1-a)}\,\mathrm{d}x \\ &= \frac{\frac{1}{2}\,a^2\,(1+b)+\frac{1}{2}\,(1-a)^2}{a(1+b)+(1-a)}\,.\end{aligned}$$

We will take the counter-example to the geometric spacing rule, the simplified Witch's Hat with parameters $a = 10^{-3}$ and $b = 10^4$, and run both the geometric scheme as well as the schedule minimising the sum of squares. Both schedules defined $t = 5$ distinct temperatures between $\beta_t = \frac{1}{16}$ and $\beta_1 = \beta_0$. In all the experiments, the results were based on 200 000 samples taken after a burn-in of 20 000 iterations. To have a point of comparison, both temperature schemes were first run under ideal world conditions. That means that the transition kernels $T_{\beta_i}(x, x')$, $i = 1, \ldots, n$ produced independent realisations of the distributions $p_{\beta_i}(x')$, $i = 1, \ldots, n$, respectively. As expected, the acceptance rates match their theoretical values closely so that the optimal scheme yielded an approximately 2.5 times higher acceptance rate than the geometric one (see Table 6-8). Comparing the autocorrelation times shows

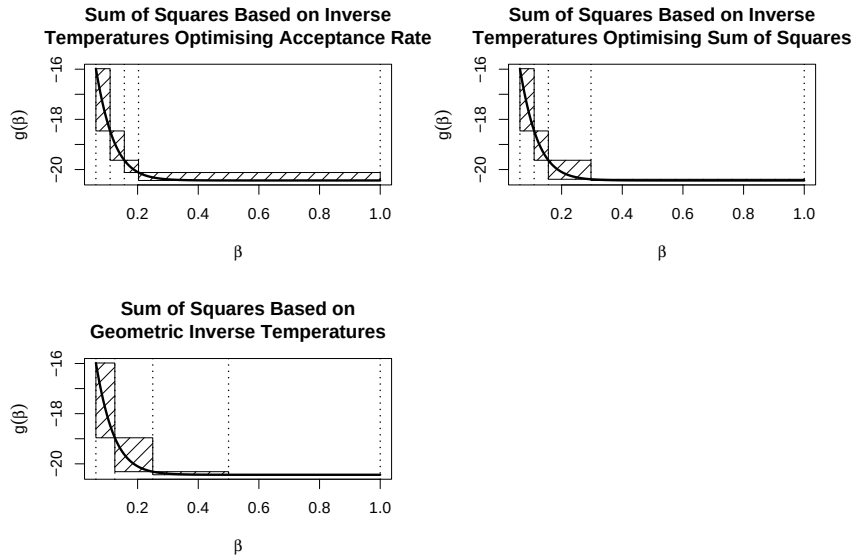

Figure 6-5: The curve $g(\beta)$ is convex when $a = 0.5$ and $b = 7.5 \cdot 10^8$. The sum of squares (shaded area) achieved by the true optimal inverse temperatures obtained by optimising the expected acceptance probability *(top left)*, by the optimal inverse temperatures obtained by optimising the sum of squares *(top right)* and by geometrically spaced inverse temperatures *(bottom left)*. In this case, the geometric scheme is quite a good choice because it places most temperatures where the curve is rapidly decaying.

that the optimal schedule is approximately three times faster in mixing. The autocorrelation times are also quite low (3.8 for the best sequence and 11.6 for the geometric rule). Since both schemes are equally expensive, the optimal scheme is clearly the better one. In a second experiment, the rapid mixing kernels were replaced by slowly mixing ones. They were defined by Metropolis updates with normal proposal $x' \sim N(x, \sigma_i^2)$ (with reflection at the boundaries of the interval $[0, 1]$) for all temperatures β_i, $i = 1, \ldots, n$, so that a step size plan σ_i, $i = 1, \ldots, n$, between the fixed σ_1 and σ_n also had to be specified. To allow local exploration at the target temperature, σ_1 was set to $\sigma_1 = a$, while $\sigma_n = 0.25$ was chosen for global exploration at the hottest temperature. The problem with step patterns is that such a plan may affect the performance of the sampler. Investigating the extent of the impact was therefore part of the experiment. In a first attempt, step sizes were chosen independent of the temperature values so that both temperature schemes were run under the same step plan. In this plan, the variances $\{\sigma_i^2\}$ grow linearly in i:

$$\sigma_i^2 = \sigma_1^2 + \frac{\sigma_n^2 - \sigma_1^2}{n}\, i, \qquad i = 1, \ldots, n. \tag{6.3}$$

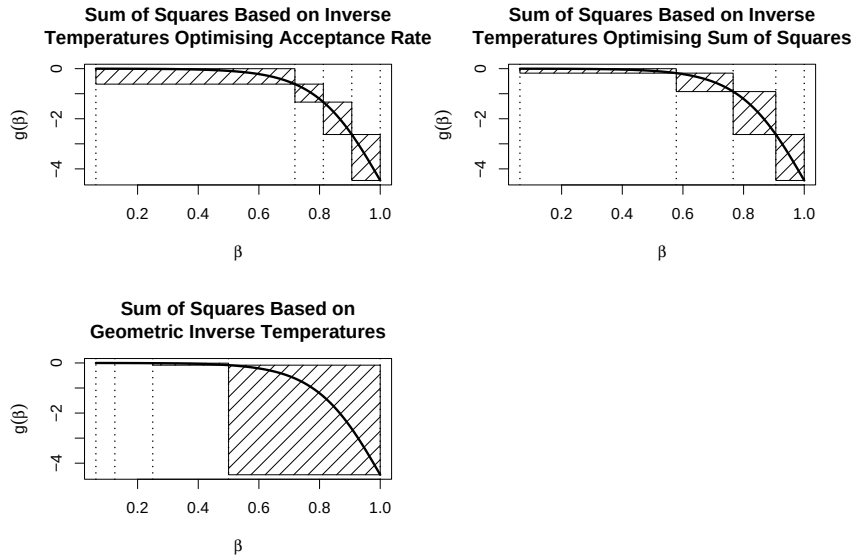

Figure 6-6: The curve $g(\beta)$ is concave when $a = 10^{-4}$ and $b = 9.5 \cdot 10^3$. The sum of squares (shaded area) achieved by the true optimal inverse temperatures obtained by optimising the expected acceptance probability *(top left)*, by the optimal inverse temperatures obtained by optimising the sum of squares *(top right)* and by geometrically spaced inverse temperatures *(bottom left)*. In this case, the geometric scheme is not a good choice because it places the temperatures where the curve is slowly decaying and not where the curve is rapidly decaying, while the optimal scheme does the inverse and thus yields a much smaller sum of squares.

The results are presented in Table 6-9. Although the optimal scheme has a higher acceptance rate than the geometric scheme, their integrated autocorrelation times do not lie that far apart: while the acceptance rate under the optimal scheme (0.267) lies clearly below the expected acceptance probability (almost half of it), the acceptance rate under the geometric scheme (0.154) is almost the same as before. The latter is more of a coincidence than a sign of good mixing because the integrated autocorrelation time is about 25 times higher than in the ideal world scenario. Another way to show the coincidence is to perform four steps at each temperature level, again under both schedules. The burn-in led to a fall in both acceptance rates. The geometric one fell to 0.098, while the optimal fell to 0.095. As in the toy example in Section 5.2.3, this drop can be explained by not many steps being carried out under the original scheme so that there is a lower proportion of proposal states that are accepted with probability one only because they are identical to the current state. Although the acceptance rates under the burn-in

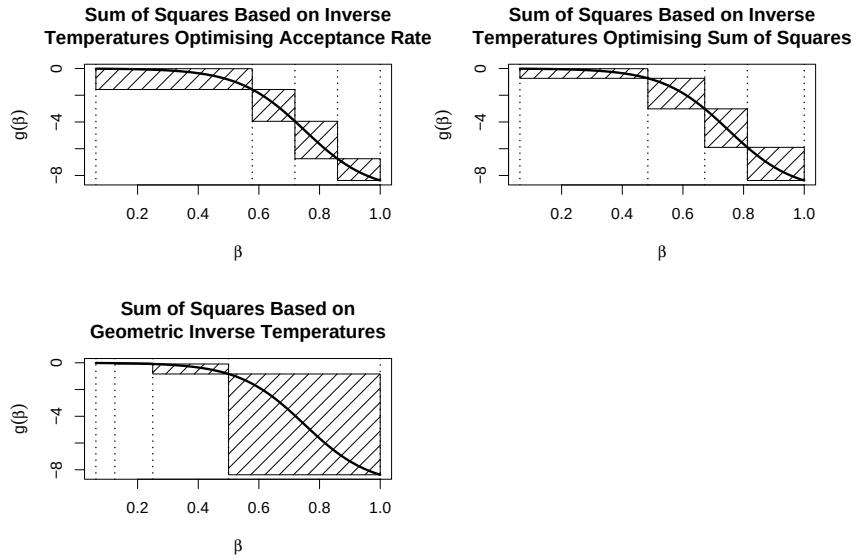

Figure 6-7: The curve $g(\beta)$ is of mixed shape when $a = 10^{-3}$ and $b = 10^4$. The sum of squares (shaded area) achieved by the true optimal inverse temperatures obtained by optimising the expected acceptance probability *(top left)*, by the optimal inverse temperatures obtained by optimising the sum of squares *(top right)* and by geometrically spaced inverse temperatures *(bottom left)*. In this case, the geometric scheme is not a good choice because it places the temperatures where the curve is slowly decaying and not where the curve is rapidly decaying, while the optimal scheme does the inverse and thus yields a much smaller sum of squares.

are the same for both temperature schedules, the mixing is not. The optimal scheme now performs about 1.5 times better than the geometric one (see Table 6-10). To vary the step pattern, a plan that depends on the temperatures was introduced:

$$\sigma_i = \sigma_1 + \frac{\sigma_n - \sigma_1}{\sqrt{1/\beta_n} - \sqrt{1/\beta_1}} \left(\sqrt{1/\beta_i} - \sqrt{1/\beta_1} \right), \qquad i = 1, 2, \ldots, n.$$

As discussed in Section 4.4.3, this choice imitates the way in which the standard deviation of a normal distribution would grow if heated. To assess the importance of the way in which the step sizes increase from σ_1 to σ_n, the plan was once defined by the temperature scheme under which it was actually run and once under the opposing scheme. This gave in total four experiments, namely optimal temperature schedule with supporting ("optimal") step plan, optimal schedule with opposing plan, geometric schedule with supporting ("geometric") plan and geometric with opposing plan. For example, if the geometric scheme was run with "optimal" step sizes, then the steps grew much

IDEAL WORLD: DIRECT DRAWS

$\{\beta_i\}$	$t \times b$	$\mathbb{E}_\varphi(\alpha)$	acceptance rate	$\hat{\tau}(x)$
geometric	5×1	0.167	0.168	11.625
optimal	5×1	0.413	0.413	3.799

Table 6-8: Ideal World: A direct draw from the tempered distribution at each temperature ($t = 5$ temperature levels, burn-in $b = 1$).

REAL WORLD: "LINEAR" STEPS

$\{\beta_i\}$	$t \times b$	acceptance rate	$\hat{\tau}(x)$
geometric	5×1	0.154	255.229
optimal	5×1	0.267	252.582

Table 6-9: Real World: Step size grow linearly in variance, but independent of temperature value ($t = 5$ temperature levels, burn-in $b = 1$).

slower than the temperatures because the optimal temperatures were much colder than the geometric ones. We have seen that, in this example, the difference between the schemes becomes clearer when we allow for a small burn-in $b = 4$ at each of the $t = 5$ distinct temperature levels. In consequence, all the experiments were run with this burn-in. The results are given in Table 6-10 together with the results of the previous burn-in experiments dealing with the independent "linear" step size plan. The step patterns are ordered according to their size at the intermediate temperatures. The "optimal" plan makes the smallest steps, while "linear" defines the largest steps. We can see that, within each group of temperature spacing, the mixing is better for the more generous step patterns "geometric" and "linear". Comparing the mixing between the two temperature groups (geometric and optimal), we find that the optimal scheme mixes more than 1.5 times faster than the geometric scheme. In this example, the temperature scheme has a greater impact on the mixing than the step size scheme. We will therefore continue concentrating more on the temperatures than on the steps.

6.6 Summary

In summary, the proposed tuning technique works quite well in the simplified Witch's Hat example. We have seen that this example is an excellent test problem for temperature schemes because the normally intractable truth is

Various Temperature and Step Size Schemes

$\{\beta_i\}$	$\{\sigma_i\}$	$t \times b$	acceptance rate	$\hat{\tau}(x)$
geometric	"optimal"	5×4	0.100	261.786
	"geometric"	5×4	0.100	153.433
	"linear"	5×4	0.098	141.364
optimal	"optimal"	5×4	0.099	113.634
	"geometric"	5×4	0.101	97.881
	"linear"	5×4	0.095	89.586

Table 6-10: Geometric and optimal temperature schemes with "optimal", "geometric" or "linear" step size plan and a small burn-in of $b = 4$ at the $t = 5$ distinct temperatures.

here tractable. The comparison between the solutions to the related and the true problem revealed that the related one yields temperatures that are close to optimal, which is very encouraging. Another important result is that the optimisation is valuable because the widely advocated geometric temperature scheme is not necessarily an efficient choice. The appropriateness can be easily inferred from the shape of $g(\beta)$: the more the behaviour differs from the model shape of $\frac{1}{2\beta}$, the less efficient is the geometric rule. The last experiments demonstrated that the optimisation results obtained in an "ideal world" scenario also much improve the results of "real world" experiments. This motivates us to investigate how the tuning technique can be applied in a complex "real world" application in the next chapter.

Chapter 7

Tempering an Applied Problem

7.1 Introduction

In this chapter, we will discuss how the temperature optimisation technique developed for tempered transitions in Chapter 5 can be applied in practice. We will test the technique on a hard applied sampling problem, namely on "label switching" in Bayesian mixture modelling which shall be explained in Section 7.2. We will demonstrate the difficulties on the example of modelling the well-known "galaxy data" by a fixed-dimensional mixture model (Section 7.3).

The first difficulty which we face when applying tempered transitions is how to temper the target distribution which is here a posterior distribution. In this example, we cannot temper the entire posterior distribution because this fully tempered distribution is improper at hot temperatures (Section 7.4). Since the likelihood contribution causes here the multimodality of the posterior distribution, we will temper the likelihood contribution only which leads to proper tempered distributions (Section 7.5). As the choice of the hottest temperature and the mixing at this temperature are crucial for the efficiency of tempered transitions, we will specify the hottest temperature and the corresponding MCMC kernel in Section 7.6. In this context, we will also test how this kernel performs at the target temperature.

The second difficulty is that we cannot choose the set of inverse temperatures $\{\beta_i\}$ minimising

$$S\left(\{\beta_i\}_{i=1}^n\right) = \sum_{i=1}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i)\right]$$

if the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ is unknown, which is the case in complex problems. We will need to estimate $g(\beta)$. We know from theory that $g(\beta)$ is decreasing. Hence, we can interpolate $g(\beta)$ by a decreasing approximation $\hat{g}(\beta)$ based on few anchor points (Section 7.7.1), which usually have to be estimated. We suggest estimating each anchor point $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ by importance sampling based on a sample $\{X_i\}$ from the hottest distribution $p_{\beta_{\min}}(x)$ from which sampling is possible (Section 7.7.2). We will also verify that such an interpolation is robust in several ways (Section 7.8). In Section 7.8, we will also discuss specifying the MCMC transition kernels used in tempered transitions. After optimising the temperatures, we will assess the performance of the tuned tempered transitions algorithm in comparison to the default tempered transitions method (Section 7.9). Finally, we will summarise the key results (Section 7.10).

7.2 Label switching in mixture modelling

Mixture models are common in many application areas such as classification or clustering (for an introduction to mixture modelling, see for example Robert 1996). In mixture modelling, we usually assume that the observations $y_1, \ldots, y_n$ are independent and come from some mixture of distributions

$$y_i \sim \sum_{k=1}^{K} w_k \, p_k(y|\theta_k), \qquad i = 1, \ldots, n.$$

The various distributions $p_k(y|\theta_k)$, $k = 1, \ldots, K$, which are specified by the corresponding parameter vector θ_k, are the "components" of the mixture, while the weights w_k, $k = 1, \ldots, K$, are the "component weights" satisfying $\sum_{k=1}^{K} w_k = 1$. It is common to define the components $p_k(\theta_k)$, $k = 1, \ldots, K$, by standard distributions. Indeed, the components come very often from the same family of distributions. For instance, univariate continuous data could be modelled by a mixture of normal distributions

$$y_i \sim \sum_{k=1}^{K} w_k \, N(\mu_k, \sigma_k), \qquad i = 1, \ldots, n, \qquad \text{independent},$$

in which case $\theta_k = (\mu_k, \sigma_k)$ and $p_k(y|\mu_k, \sigma_k) = N(\mu_k, \sigma_k)$. The simplest Bayesian mixture model defines independent prior distributions $p(\{w_k\})$ and $p(\{\theta_k\})$ for the component weights and the component parameters so that the posterior distribution takes the following form:

$$p\left(\{w_k, \theta_k\} \,\middle|\, \{y_i\}\right) \propto p(\{w_k\})\, p(\{\theta_k\}) \prod_{i=1}^{n} \sum_{k=1}^{K} w_k \, p_k(y_i|\theta_k).$$

It is also possible to define a more complex hierarchical model which may include hyperparameters for the component parameters or allocation variables $z_1, \ldots, z_n$ for the n observations. The variables $z_1, \ldots, z_n$ allocate each of the observations $y_1, \ldots, y_n$ to one of the components $p_k(y|\theta_k)$, $k = 1, \ldots, K$, so that we can write

$$y_i|z_i \sim p_{z_i}(y|\theta_{z_i}, z_i), \qquad i = 1, \ldots, n.$$

Each z_i takes a particular value k, $k = 1, \ldots, K$, with prior probability w_k. This defines a simple hierarchical Bayesian model with independent priors for component weights and component parameters and conditional prior $p\left(\{z_k\} \,\middle|\, \{w_k\}\right)$ for the allocation variables

$$p\left(\{z_k, w_k, \theta_k\} \,|\, \{y_i\}\right) \propto p\left(\{w_k\}\right) p\left(\{\theta_k\}\right) p\left(\{z_i\} \,|\, \{w_k\}\right) \prod_{i=1}^{n} p_{z_i}(y_i|\theta_{z_i}, z_i).$$

We usually choose conjugate priors so that we can sample from this model by Gibbs sampling. Indeed, the possibility of Gibbs sampling is often the reason for introducing allocation variables.

It is easiest to explain the "label switching" problem in mixture modelling when the components $p_k(y|\theta_k)$, $k = 1, \ldots, K$, come all from the same family of distributions (e.g. from the family of univariate normal distributions) and when there are no allocation variables, i.e. in the case that

$$p\left(\{w_k, \theta_k\} \,|\, \{y_i\}\right) \propto p\left(\{w_k\}\right) p\left(\{\theta_k\}\right) \prod_{i=1}^{n} \sum_{k=1}^{K} w_k \, p_k(y_i|\theta_k).$$

Due to the commutativity of summands, the mixture model is invariant to permutation of the labels so that, for any permutation $\rho(k)$, the following model

$$p\left(\{w_{\rho(k)}, \theta_{\rho(k)}\} \,|\, \{y_i\}\right) \propto p\left(\{w_{\rho(k)}\}\right) p\left(\{\theta_{\rho(k)}\}\right) \prod_{i=1}^{n} \sum_{k=1}^{K} w_{\rho(k)} \, p_{\rho(k)}\left(y_i|\theta_{\rho(k)}\right)$$

is equivalent to the original model. If there is no rule for labelling the components, the sampler should visit all the $K!$ possible permutations when sampling from the posterior distribution. This may become clearer in the simple example of modelling a mixture of two normal distributions

$$w_1 N(\mu_1, \sigma_1^2) + w_2 N(\mu_2, \sigma_2^2)\,.$$

Suppose the expected "true" model is

$$\frac{1}{4}\, N(78, 5^2) + \frac{3}{4}\, N(6, 2^2)\,.$$

If we permute the components, we obtain the equivalent model

$$\frac{3}{4}\,N(6,2^2)+\frac{1}{4}\,N(78,5^2)\,.$$

As all permutations are equally likely, we should observe that our sampled values for (w_1,μ_1,σ_1^2) spend an equal amount of time close to $(\frac{1}{4},78,5^2)$ and close to $(\frac{3}{4},6,2^2)$, while (w_2,μ_2,σ_2^2) should swap accordingly between $(\frac{3}{4},6,2^2)$ and $(\frac{1}{4},78,5^2)$. Each permutation represents a mode of the posterior distribution so that moving between these permutations ("label switching") is a way of mode jumping. Standard MCMC methods often have difficulties achieving label switching. A lack of label switching can be diagnosed by comparing the histograms of the sampled component values. If the sampler has converged, all the histograms of one kind should be similar since the marginal posterior distributions for each type of random variable (such as component weight, mean and variance) are theoretically identical. For instance, if we model a mixture of normal distributions, then the histograms of the weights $w_k,\ k=1,\ldots,K$, should resemble each other; similarly, the histograms of the component means $\mu_k,\ k=1,\ldots,K$, should look the same, and the histograms of the variances $\sigma_k^2,\ k=1,\ldots,K$, should also appear identical. If the histograms do not show such a symmetry, strictly speaking, convergence has not taken place. Label switching can be achieved trivially, for example by a permutation move that updates components j and k by allocating the previous parameter values of component j to component k and vice versa. We are however not interested in this trivial kind of label switching because it does not provide any information whether the sampler is in general able to mix between difficult parts of the sample space without explicit mode jumping directions. If, on the other hand, a sampler achieves label switching without direct instructions, then it can jump between modes that are hard to attain and we can be fairly confident that the sampler mixes also well elsewhere. As label switching is easy to monitor, but hard to achieve, it is an excellent test problem for mode jumping in MCMC.

7.3 A model for the galaxy data

7.3.1 Galaxy data

As an applied problem, we will test sampling from a Bayesian mixture model for the well-known "galaxy data". The galaxy data are astrophysical data consisting of the velocities at which 82 galaxies in the Corona Borealis region

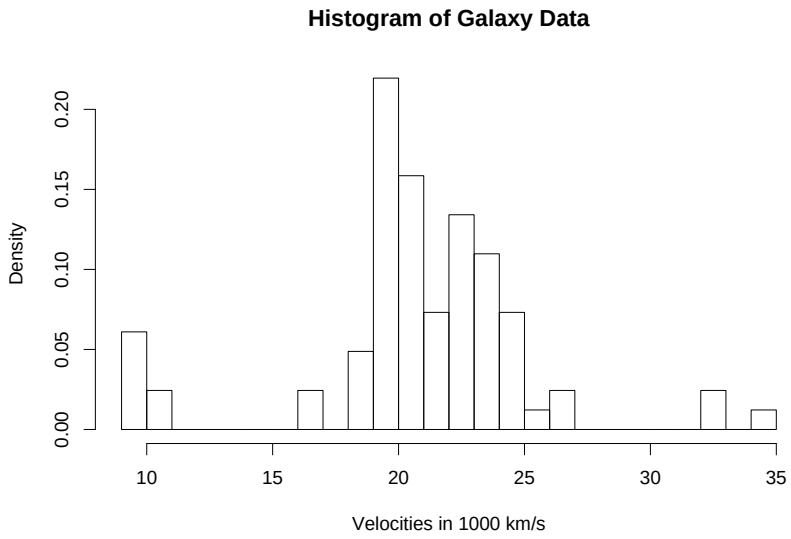

Figure 7-1: The histogram of the "galaxy" data indicates that the underlying distribution is multimodal.

are moving away from our galaxy. Analysing these data can give evidence for voids and superclusters in the universe. The galaxy data were introduced into the statistical literature by Roeder (1990) where they can also be found in one of the tables. Alternatively, the galaxy data can be accessed in R (version 2.4.0) by loading the R library MASS and calling `galaxies`. Note that there are transposed digits in the 78th galaxy observation in R: the 78th observation is `galaxies[78]` = `26690` (kilometres per second), although it should read `galaxies[78]` = `26960` as given in Roeder (1990). This error is acknowledged in the documentation of the R version 2.4.0. A histogram of the galaxy data can be found in Figure 7-1.

7.3.2 Richardson and Green's model

We will develop a Bayesian model based on Richardson and Green's (1997) approach. This approach models the data by a mixture of univariate normal distributions and allows the number of components to vary because one interesting aspect of the galaxy problem is to determine the most likely number of components. For completeness, we will present the variable-component model although we will fix the number of components later in this chapter to test mode jumping in fixed dimension. Richardson and Green's model is a hierarchical model containing allocation variables and hyperparameters. In this

model, the observations $x_1, \ldots, x_n$ are independent realisations of a mixture of normal distributions

$$y_i \sim \sum_{k=1}^{K} w_k \, N(\mu_k, \sigma_k), \qquad i = 1, \ldots, n, \qquad \text{independent.}$$

The allocation variables $z_1, \ldots, z_n$ assign the observations $x_1, \ldots, x_n$ to the kth component with prior probability w_k so that

$$\begin{aligned} y_i | z_i &\sim N(\mu_{z_i}, \sigma_{z_i}), && i = 1, \ldots, n, \\ p(z_i = k) &= w_k, && i = 1, \ldots, n. \end{aligned}$$

The prior distribution for the number K of components is a uniform distribution on the possible number of components $\{1, \ldots, K_{\max}\}$. Conditional on K, the component weights $\{w_k\}$ have a Dirichlet distribution as prior, while the component means $\{\mu_k\}$ and variances $\{\sigma_k^2\}$ are drawn independently from normal and gamma priors so that

$$\begin{aligned} K &\sim U\{1, \ldots, K_{\max}\} \\ \{w_k\} | K &\sim \textit{Dirichlet} \underbrace{(\delta, \ldots, \delta)}_{K \text{ times}} \\ \mu_k | K &\sim N(\xi, \kappa^{-1}), && k = 1, \ldots, K, \\ \sigma_k^2 | K &\sim \textit{Inverse Gamma}(\alpha, \beta), && k = 1, \ldots, K. \end{aligned}$$

Richardson and Green (1997) fix most of the hyperparameters. They want the prior $N(\xi, \kappa^{-1})$ to be flat over an interval of variation of the data so that they choose ξ to be the midpoint of the interval while setting κ to $\kappa = 1/R^2$, where R is the length of the interval. Similarly, the Dirichlet parameter $\delta = 1$ is fixed. The only hyperparameter that is variable is β. The β variable is assumed to follow a gamma distribution with parameters g and h,

$$\beta \sim \textit{Gamma}(g, h),$$

where the scale parameter h is set to $h = 10/R^2$. The remaining hyperparameters α (from $\sigma_k^2 | K \sim \textit{Inverse Gamma}(\alpha, \beta)$) and g (from $\beta \sim \textit{Gamma}(g, h)$) are chosen such that $\alpha > 1 > g$, namely $\alpha = 2$ and $g = 0.2$, to reflect the prior belief that the component variances $\{\sigma_k^2\}$ are of a similar size. Richardson and Green (1997) state that all these prior distributions have the advantage of conjugacy, but that this advantage is not necessarily needed in MCMC computation. In their implementation, they are able to use a Gibbs kernel for updating all these variables. Richardson and Green (1997) avoid the label switching problem by imposing an ordering constraint on the labels requiring that the components are always labelled such that the component means are ordered in increasing order so that $\mu_1 < \mu_2 < \cdots < \mu_K$. Celeux, Hurn

and Robert (2000) warn against employing identifiability constraints in the sampling stage because these tend to truncate the unconstrained posterior distribution without respecting its geometry and shape. The chances are that the ordering constraint does not nicely separate the $K!$ permutation modes of the unconstrained distribution, but that it will define a truncation involving parts of several modal regions. This also has consequences for inference. For example, the posterior mean would then lie somewhere between the $K!$ modes rather than close to one of these modes as intended. As a remedy, Celeux et al. (2000) use the unconstrained posterior distribution for sampling from mixture models. In their examples, which do not include the galaxy data, they employ tempered transitions to achieve label switching. Furthermore, they avoid improper tempered distributions by tempering the likelihood contribution of the posterior while leaving the prior part unchanged. They choose a geometric temperature scheme and report low acceptance rates. For statistical inference, they use a decision-theoretic approach based on label-invariant loss functions so that there is no need to find a suitable way of labelling. Note that Jasra, Holmes and Stephens (2005) also use tempered transitions to sample from mixture models. In particular, they investigate sampling from Richardson and Green's galaxy model when the number of components is fixed at $K = 6$. In contrast to Richardson and Green, they do not impose any ordering constraints. They do not attempt tempering the full posterior because they expected to obtain very small acceptance rates when tempering the full posterior, which was the case in a different example they investigated. They choose instead to temper the full conditional distributions $\left[\text{e.g. } p\left(\{\mu_k\} \,\middle|\, \cdots\right) \text{ and } p\left(\{\sigma_k^2\} \,\middle|\, \cdots\right)\right]$. They do not comment on the temperature scheme they employed. In their summary, they note in general that they had problems tuning the tempering sampler whenever there were highly separated modes in the posterior distribution. These reported difficulties when sampling from Bayesian mixture models by tempered transitions also motivate the following work on optimising tempered transitions in mixture modelling.

7.3.3 Final model for galaxy data

Richardson and Green's (1997) approach is quite complex and requires trans-dimensional MCMC methods due to the variable number K of mixture components. It is worth simplifying this model here because we are not so much interested in the statistical inference, but in the hard sampling problem caused by the non-identifiability of labels. A first step towards simplification

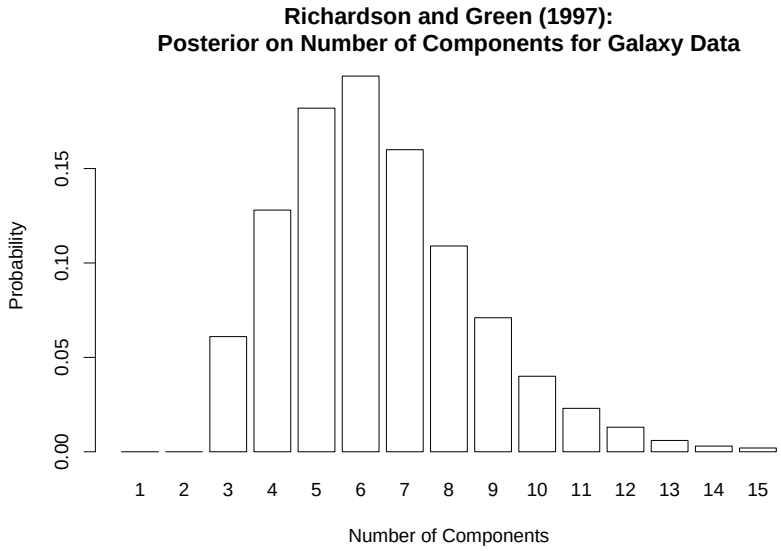

Figure 7-2: The plot shows the posterior probabilities for the number of components under the "galaxy" model that Richardson and Green (1997) analyse. The probabilities are taken from their Table 1.

is to fix the number K of components. Looking at Richardson and Green's (1997) results (see Figure 7-2), at least $K = 3$ components are needed to explain the galaxy data so that $K = 3$ seems a sensible choice. It is also the choice at which label switching is hardest because there is no "free" component, i.e. a component of tiny weight, that can be used for label switching. Free components help swap labels for, if the free component lies close to an important component, a weight update can easily reverse the importance of the components so that the previously trapped component can freely travel to a third important component and swap roles there by another weight update. Apart from fixing K in the galaxy model, we will also discard the allocation variables and keep the hyperparameters constant. As in Richardson and Green (1997), we will choose vague priors for component weights $\{w_k\}$, means $\{\mu_k\}$ and variances $\{\sigma_k^2\}$. But we will not impose any identifiability constraints on the labelling of components because label switching is what we want to test. To simplify the dealing with indices, we will use the short notation

$$\underline{n} := \{1, 2, \ldots, n\}$$

where n can be any positive integer so that we can also denote

$$\underline{(K-1)} = \{1, 2, \ldots, K-1\}$$

etc. Using the short notation, the final fixed component model for the galaxy data y_i, $i = 1, 2, \ldots, n$, is defined by

$$\begin{aligned} y_j \Big| \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}} &\sim \textstyle\sum_{k=1}^{K} w_k \, N(\mu_k, \sigma_k^2), && \forall\, j \in \underline{n}, \\ \{w_k\}_{k \in \underline{K}} &\sim \mathit{Dirichlet}(1, \ldots, 1), \\ \mu_k &\sim N(0, 1000), && \forall\, k \in \underline{K}, \\ \sigma_k^2 &\sim \mathit{Inverse\ Gamma}(1, 1), && \forall\, k \in \underline{K}. \end{aligned}$$

Note that the marginal distribution of a particular weight w_k is $Beta(1, K-1)$, i.e. $p(w_k) = (K-1)\,(1-w_k)^{K-2}$ for $0 \leq w_k \leq 1$. The likelihood functions are

$$p\left(y_j \Big| \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right) \propto \sum_{k=1}^{K} w_k \, \left(\sigma_k^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_k^2}\,(y_j - \mu_k)^2\right], \qquad \forall\, j \in \underline{n},$$

while the prior distributions are

$$\begin{aligned} p\left(\{w_k\}_{k \in \underline{K}}\right) = &\quad (K-1)!\, \mathbb{1}_{\left\{\sum_{k=1}^{K} w_k = 1\right\}}, && w_k \in [0,1], && \forall\, k \in \underline{K}, \\ p\left(\mu_k\right) = &\quad \left(2 \cdot 1000\,\pi\right)^{-\frac{1}{2}} \exp\left(-\tfrac{\mu_k^2}{2 \cdot 1000}\right), && \mu_k \in (-\infty, \infty), && \forall\, k \in \underline{K}, \\ p\left(\sigma_k\right) = &\quad \left(\sigma_k\right)^{-2} \exp\left(-\tfrac{1}{\sigma_k^2}\right), && \sigma_k^2 \in (0, \infty), && \forall\, k \in \underline{K}, \end{aligned}$$

Note that w_K is a dummy variable as it can be written by

$$w_K = 1 - \sum_{k=1}^{K-1} w_k.$$

This implies that the dimension of the K-component model is $(3K-1)$. It also means that we do not integrate over the dummy variable w_K when integrating over the prior or posterior distribution. Note also that the component means and variances are a-priori independent of each other and independent of the weights, while the weights depend on each other. The joint prior distributions are thus of the following product form:

$$\begin{aligned} p\left(\{\mu_k\}_{k \in \underline{K}}\right) &= \prod_{k=1}^{K} p\left(\mu_k\right), \\ p\left(\left\{\sigma_k^2\right\}_{k \in \underline{K}}\right) &= \prod_{k=1}^{K} p\left(\sigma_k^2\right), \\ p\left(\{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right) &= p\left(\{w_k\}_{k \in \underline{K}}\right)\, p\left(\{\mu_k\}_{k \in \underline{K}}\right)\, p\left(\left\{\sigma_k^2\right\}_{k \in \underline{K}}\right). \end{aligned}$$

As the data are also conditionally independent, the joint likelihood function is given by

$$p\left(\{y_j\}_{j \in \underline{n}} \Big| \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right) = \prod_{j=1}^{n} p\left(y_j \Big| \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right)$$

so that the posterior distribution is then

$$p\left(\{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}} \Big| \{y_j\}_{j \in \underline{n}}\right) \propto p\left(\{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right)\, p\left(\{y_j\}_{j \in \underline{n}} \Big| \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right).$$

7.4 Improper tempering - a cautionary example

The posterior for the final galaxy model is a good example to show that we have to be careful when defining tempered versions of the posterior. Tempering the full posterior leads here to improper distributions at hot (small) inverse temperatures β. We will show that the fully tempered posterior

$$\begin{aligned} & p_\beta\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}} \middle| \left\{y_j\right\}_{j \in \underline{n}}\right) \\ & \quad \propto \left[p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right)\right]^\beta \left[p\left(\left\{y_j\right\}_{j \in \underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right)\right]^\beta \\ & \quad \propto \left[p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right)\right]^\beta \prod_{j=1}^n \left\{\sum_{k=1}^K w_k \left(\sigma_k^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_k^2}\left(y_j - \mu_k\right)^2\right]\right\}^\beta, \end{aligned}$$

is improper for $\beta < \frac{2}{n+4}$. The proof relies on the inequality

$$\sum_{k=1}^K w_k \left(\sigma_k^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_k^2}\left(y_j - \mu_k\right)^2\right] \geq w_K \left(\sigma_K^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_K^2}\left(y_j - \mu_K\right)^2\right], \tag{7.1}$$

which holds because all the summands on the left hand side are non-negative. For brevity, we will denote the full vector of random variables by θ, and the vector comprising all random variables apart from σ_K^2 by $\theta_{-\sigma_K^2}$ so that

$$\begin{aligned} \theta &= \left(\left\{w_k\right\}_{k \in \underline{(K-1)}}, \left\{\mu_k\right\}_{k \in \underline{K}}, \left\{\sigma_k^2\right\}_{k \in \underline{K}}\right) \\ \text{and} \qquad \theta_{-\sigma_K^2} &= \left(\left\{w_k\right\}_{k \in \underline{(K-1)}}, \left\{\mu_k\right\}_{k \in \underline{K}}, \left\{\sigma_k^2\right\}_{k \in \underline{(K-1)}}\right). \end{aligned}$$

Note that θ does not include the dummy variable w_K as $w_K = 1 - \sum_{k=1}^{K-1} w_k$ is a function of the random variables $\left\{w_k\right\}_{k \in \underline{(K-1)}}$. If we integrate over the fully tempered distribution for $\beta < \frac{2}{n+4}$

$$\begin{aligned} & \int \mathrm{d}\theta\, p_\beta\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}} \middle| \left\{y_j\right\}_{j \in \underline{n}}\right) \\ & \quad \propto \int \mathrm{d}\theta \left[p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right)\right]^\beta \prod_{j=1}^n \left\{\sum_{k=1}^K w_k \left(\sigma_k^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_k^2}\left(y_j - \mu_k\right)^2\right]\right\}^\beta \\ & \quad \overset{(7.1)}{\geq} \int \mathrm{d}\theta \left[p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right)\right]^\beta \prod_{j=1}^n \left\{w_K \left(\sigma_K^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_K^2}\left(y_j - \mu_K\right)^2\right]\right\}^\beta \\ & \quad = \int \mathrm{d}\theta_{-\sigma_K^2} \left[p\left(\left\{w_k\right\}_{k \in \underline{K}} \left\{\mu_k\right\}_{k \in \underline{K}} \left\{\sigma_k^2\right\}_{k \in \underline{(K-1)}}\right)\right]^\beta w_K^{n\,\beta} \\ & \qquad \int_0^\infty \mathrm{d}\sigma_K^2 \underbrace{\left(\sigma_K^2\right)^{-\frac{n+4}{2}\beta} \exp\left\{-\frac{\beta}{\sigma_K^2}\left[1 + \frac{1}{2}\sum_{j=1}^n (y_j - \mu_K)^2\right]\right\}}_{\propto\ Inverse\ \ Gamma\left(\frac{n+4}{2}\beta - 1,\, \beta\left[1 + \frac{1}{2}\sum_{j=1}^n (y_j - \mu)^2\right]\right)} \\ & \quad = \infty, \end{aligned} \tag{7.2}$$

we obtain infinity because the parameter $\left(\frac{n+4}{2}\beta - 1\right)$ of the inverse gamma distribution is negative by the choice of β so that the inverse gamma

distribution is improper and integrates to infinity. We have shown that the fully tempered posterior is improper for $\beta \in \left(-\infty, \frac{2}{n+4}\right)$, but this does not imply that the distribution is proper for all $\beta \in \left[\frac{2}{n+4}, 1\right)$. If we want to base tempered transitions on inverse temperatures in $\left[\frac{2}{n+4}, 1\right)$, we have to prove that these temperatures define proper distributions. There may be a range of β values for which the fully tempered posterior distribution is proper, but this range cannot be easily determined. It might be easier to choose an alternative way of tempering as we will suggest in the next section.

7.5 Proper tempered distributions

As the prior distributions for the galaxy mixture model are independent and unimodal, the multimodality of the posterior distribution

$$p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}} \middle| \left\{y_j\right\}_{j\in\underline{n}}\right) \propto p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right) p\left(\left\{y_j\right\}_{j\in\underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)$$

is caused solely by the likelihood $p\left(\left\{y_j\right\}_{j\in\underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)$. To help the mixing of the sampler, it is therefore sufficient to temper the likelihood part only as suggested in Celeux et al. (2000). The prior part $p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)$ is left unchanged. The partly tempered posterior distribution is then

$$p_\beta\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}} \middle| \left\{y_j\right\}_{j\in\underline{n}}\right) \propto p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right) \left[p\left(\left\{y_j\right\}_{j\in\underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\right]^\beta.$$

Following previous notation, we will write

$$p_\beta\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}} \middle| \left\{y_j\right\}_{j\in\underline{n}}\right) \propto p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right) \exp\left[-\beta\, h\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\right]$$

where

$$h\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right) = -\log\left[p\left(\left\{y_j\right\}_{j\in\underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\right]$$

is the energy function. This definition gives proper tempered distributions when $\beta \in (0, 1)$ as we will show below. In general, tempering only the likelihood contribution always defines a proper tempered posterior distribution if $\beta \in (0, 1)$ provided that the prior and posterior are proper distributions. This can be verified by Hölder's inequality. Hölder's inequality states that if $p > 1$ and $q > 1$ are real numbers such that $\frac{1}{p} + \frac{1}{q} = 1$, then for any measurable functions $f, g : \Omega \to \{-\infty\} \cup \mathbb{R} \cup \{\infty\}$ the following inequality holds:

$$\int |f\, g|\, \mathrm{d}\mu \leq \left(\int |f|^p\, \mathrm{d}\mu\right)^{1/p} \left(\int |g|^q\, \mathrm{d}\mu\right)^{1/q}$$

(see for example Bauer 2001, Theorem 14.1). From this inequality it follows that the tempered distribution $p_\beta(\theta|x) \propto p(\theta)\ [p(x|\theta)]^\beta$ is proper for any $\beta \in$

$(0,1)$ provided that the prior $p(\theta)$ and the posterior $p_\beta(\theta|x)$ are proper, which means that $\int p(\theta)\mathrm{d}\theta < \infty$ and $\int p(\theta)\, p(x|\theta)\mathrm{d}\theta < \infty$, as then

$$\begin{aligned}\int p(\theta)\ [p(x|\theta)]^\beta \,\mathrm{d}\theta &= \int \left|[p(\theta)]^{(1-\beta)}\ [p(\theta)\, p(x|\theta)]^\beta\right| \mathrm{d}\theta \\ &< \left(\int \left|[p(\theta)]^{(1-\beta)}\right|^{1/(1-\beta)} \mathrm{d}\mu\right)^{(1-\beta)} \left(\int \left|[p(\theta)\, p(x|\theta)]^\beta\right|^{1/\beta} \mathrm{d}\mu\right)^{\beta} \\ &= \left(\int p(\theta)\mathrm{d}\mu\right)^{(1-\beta)} \left(\int p(\theta)\, p(x|\theta)\mathrm{d}\mu\right)^{\beta} \\ &< \infty.\end{aligned}$$

7.6 Sampling from the hottest and the coldest distribution by standard MCMC

Having found a valid way of tempering, namely

$$p_\beta\left(\left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}} \middle|\, \{y_j\}_{j\in\underline{n}}\right) \propto p\left(\left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}}\right) \exp\left[-\beta\, h\left(\left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}}\right)\right]$$

where

$$h\left(\left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}}\right) = -\log\left[p\left(\{y_j\}_{j\in\underline{n}} \middle|\, \left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}}\right)\right],$$

we now need to find a sufficiently hot inverse temperature $\beta_{\min}$ to allow label switching when sampling from $p_{\beta_{\min}}\left(\left\{w_k,\mu_k,\sigma_k^2\right\}_{k\in\underline{K}} \middle|\, \{y_j\}_{j\in\underline{n}}\right)$, $K=3$, by standard MCMC. The hottest inverse temperature $\beta_{\min}$ is a crucial parameter in tempered transitions. On the one hand, we know that $\beta_{\min}$ should be chosen sufficiently small so that standard MCMC mixes well between modes of the hottest distribution. On the other hand, we have seen that $\beta_{\min}$ should not be set much smaller than necessary since the smaller $\beta_{\min}$ is chosen, the more intermediate temperature levels are required to give reasonable acceptance rates. Here the values $\beta_{\min} = 2^{-i}$, $i = 2, 4, 8, 16$, were considered. It was found that $\beta_{\min} = \frac{1}{8}$ satisfied the above criterion. While the marginal posterior distributions of w_1, w_2, w_3 and of σ_1^2, σ_2^2, σ_3^2 seem unimodal (see histograms in Figures 7-3 and 7-7), the modes of the marginal posteriors of μ_1, μ_2, μ_3 just touch (see histogram in Figure 7-5). The traceplots and histograms (Figures 7-3, 7-5 and 7-7) show that the algorithm is mixing well between labels. Note that the trace plots are thinned; they show every 100th sample. The histograms of w_1, w_2, w_3 resemble each other as do the histograms of μ_1, μ_2, μ_3 and σ_1^2, σ_2^2, σ_3^2 so that convergence of the algorithm can be inferred.

Before describing the MCMC sampler used to generate these samples, we will introduce a measure for the quality of label switching. Such a measure is needed

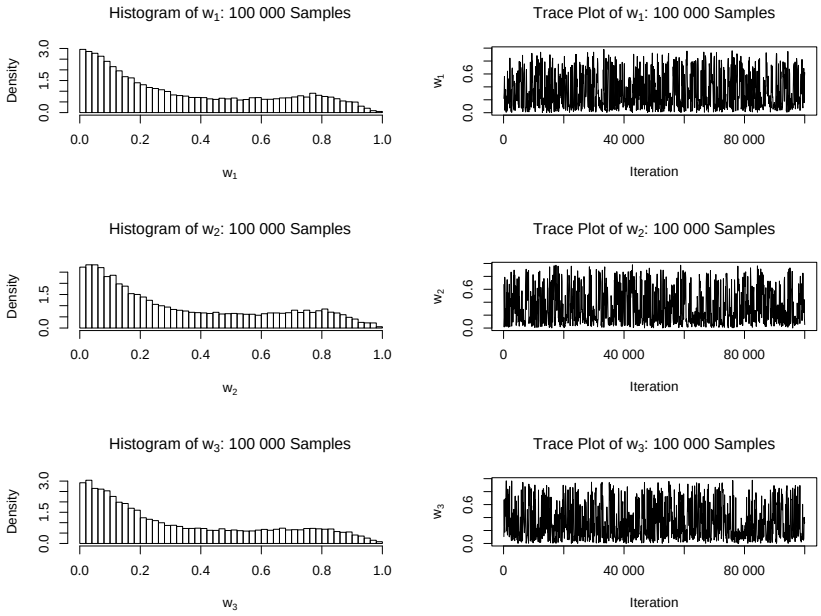

Figure 7-3: The histograms and traceplots of the posterior weights w_1, w_2, w_3 at the hottest temperature $\beta_{\min} = \frac{1}{8}$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

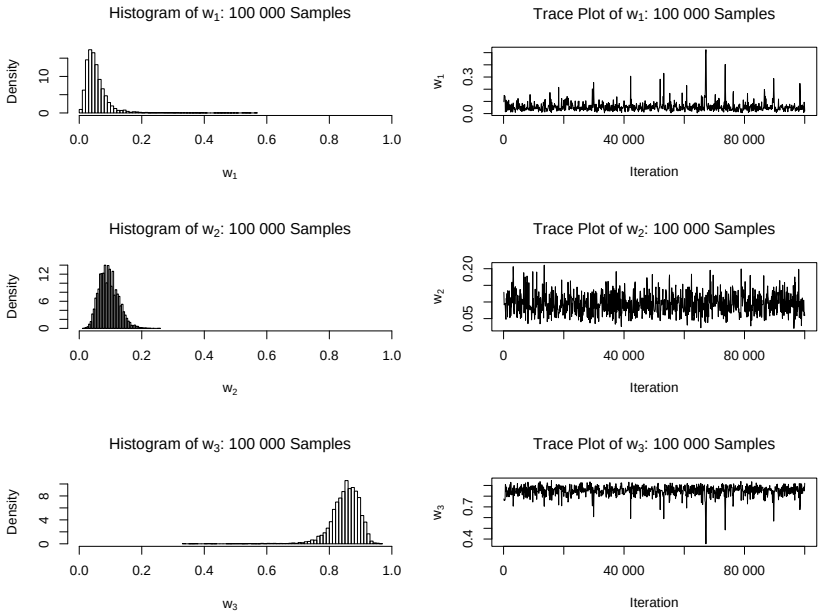

Figure 7-4: The histograms and traceplots of the posterior weights w_1, w_2, w_3 at the target temperature $\beta_0 = 1$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The asymmetry between histograms indicates the lack of label switching.

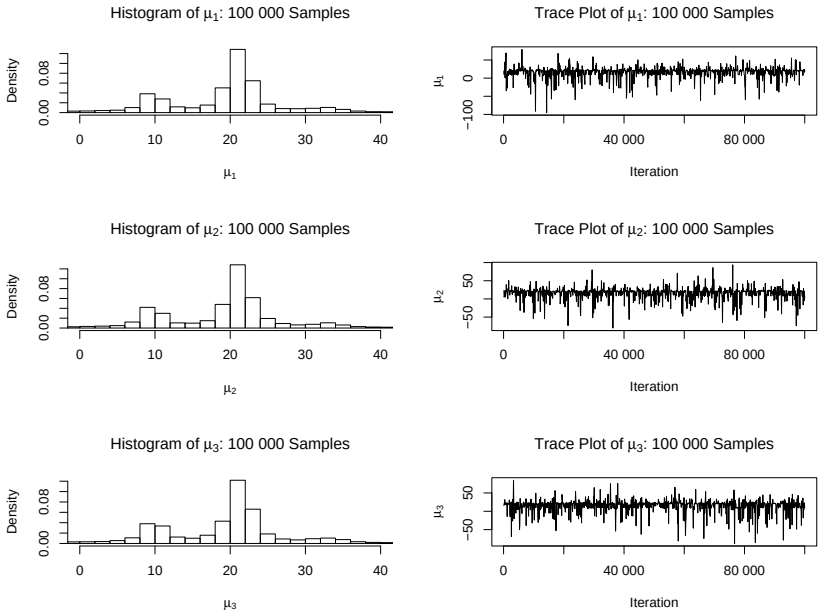

Figure 7-5: The histograms and traceplots of the posterior means μ_1, μ_2, μ_3 at the hottest temperature $\beta_{\min} = \frac{1}{8}$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

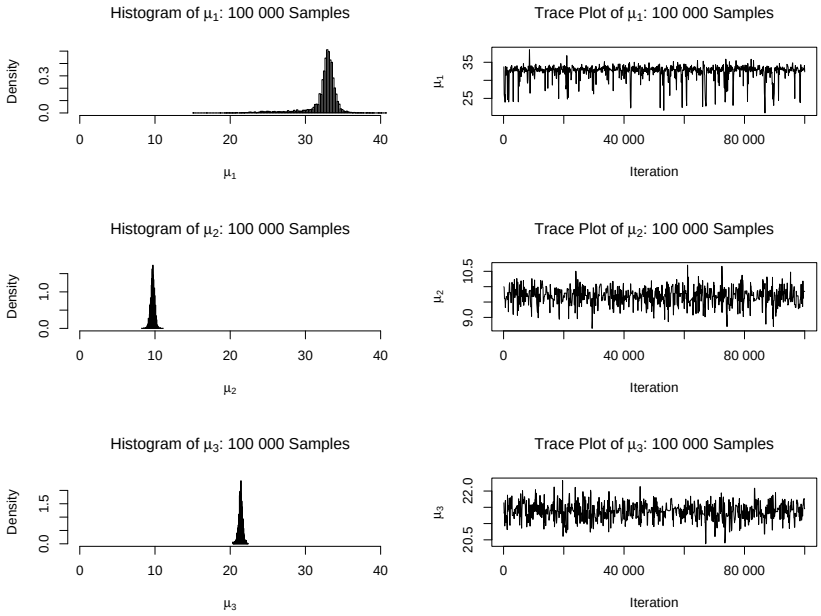

Figure 7-6: The histograms and traceplots of the posterior means μ_1, μ_2, μ_3 at the target temperature $\beta_0 = 1$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The asymmetry between histograms indicates the lack of label switching.

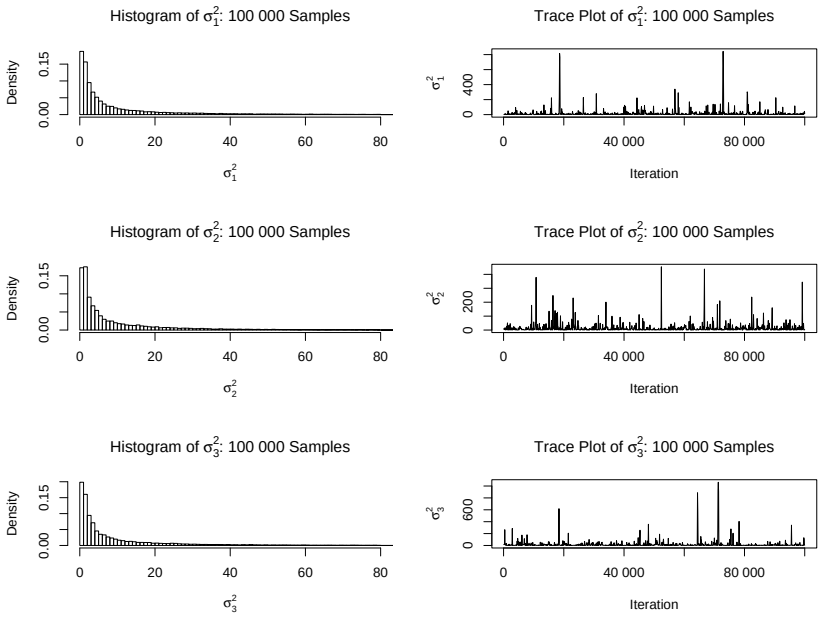

Figure 7-7: The histograms and traceplots of the posterior variances σ_1^2, σ_2^2, σ_3^2 at the hottest temperature $\beta_{\min} = \frac{1}{8}$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

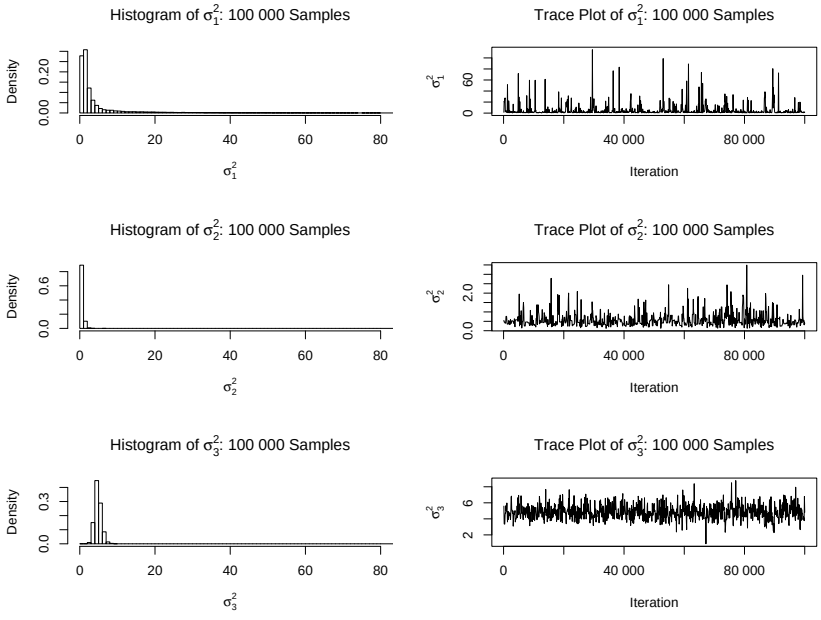

Figure 7-8: The histograms and traceplots of the posterior variances σ_1^2, σ_2^2, σ_3^2 at the target temperature $\beta_0 = 1$ obtained by standard MCMC (100 000 iterations). The traceplots show every 100th sample. The asymmetry between histograms indicates the lack of label switching.

to decide on efficient step sizes for the MCMC updates. To measure the mixing between labels, we can estimate the integrated autocorrelation time (2.3) for each random variable with respect to the corresponding "group mean" of all the random variables of the same type. Let us explain this concept for the example of calculating the integrated autocorrelation time $\tau(\mu_k)$ for each $k = 1, \ldots, K$. Due to the non-identifiability of labels, $\mu_1, \ldots, \mu_K$ are theoretically identically distributed. Let us denote the common marginal distribution by $\psi(\mu)$ so that $\mu_k \sim \psi(\mu)$, $k = 1, \ldots, K$. In consequence, the integrated autocorrelation time $\tau(\mu_k)$ is taken with respect to the theoretical mean $\mathbb{E}_\psi(\mu)$. To estimate the integrated autocorrelation time $\tau(\mu_k)$, we thus have to estimate $\mathbb{E}_\psi(\mu)$. If we base the estimate for $\mathbb{E}_\psi(\mu)$ solely on the μ_k-samples $\left\{\mu_k^{(t)}\right\}_{t=1}^{N}$, then we will obtain a poor estimate if the sampler does not mix well between labels and gets stuck in one of the permutation modes. If we estimate $\mathbb{E}_\psi(\mu)$ by the pooled sample mean

$$\bar{\mu} = \frac{1}{K}\sum_{k=1}^{K}\bar{\mu}_k,$$

we should obtain good estimates even if the MCMC sampler is not able to switch labels. We can then estimate the integrated autocorrelation times $\hat{\tau}(\mu_k)$, $k = 1, \ldots, K$, each with respect to the pooled mean $\bar{\mu}$. The pooled mean not only gives a more accurate τ estimate, it also detects the lack of convergence when the sampler does not mix between labels; for then the integrated autocorrelation time estimates do not converge, which can be recognised by the window width of the estimator (2.3) being as large as the sample size. The same ideas apply to calculating τ for the weights and variances so that we will estimate the corresponding integrated autocorrelation times $\hat{\tau}(w_k)$ and $\hat{\tau}(\sigma_k^2)$, $k = 1, \ldots, K$, with respect to the pooled means $\bar{w} = \frac{1}{K}\sum_{k=1}^{K}\bar{w}_k$ and $\bar{\sigma}^2 = \frac{1}{K}\sum_{k=1}^{K}\bar{\sigma}_k^2$ respectively. The autocorrelation times are used to tune the step sizes of the standard MCMC sampler for the hottest distribution.

When sampling from the hottest distribution, there are three types of random variables that need updating, the component means $\{\mu_k\}_{k\in\underline{K}}$, the component variances $\{\sigma_k^2\}_{k\in\underline{K}}$ and the component weights $\{w_k\}_{k\in\underline{K}}$. The component means and variances were updated one by one. For updating the component means, a Metropolis algorithm with normally distributed proposals

$$\mu_k' \sim N\left(\mu_k, \sigma_\mu^2\right)$$

was chosen. By trial and error, the step size was set to $\sigma_\mu = 25.0$ because the resulting estimated integrated autocorrelation times $\hat{\tau}(\mu_k)$, $k = 1, 2, 3$, were

smaller than the ones obtained by setting $\sigma_\mu = 20.0$ or $\sigma_\mu = 30.0$. The chosen proposal mechanism for updating the component variances reflects that the variances are non-negative: if σ_k^2 is the current state, a proposal state $(\sigma_k')^2$ is created by first drawing $z \sim N(0, \sigma_\sigma^2)$ and then setting

$$(\sigma_k')^2 = \begin{cases} \sigma_k^2 + z & \text{if } \sigma_k^2 + z \geq 0 \\ -(\sigma_k^2 + z) & \text{otherwise.} \end{cases}$$

This proposal is symmetric as

$$q\left(\sigma_k^2, (\sigma_k')^2\right) = q_Z\left((\sigma_k')^2 - \sigma_k^2\right) + q_Z\left((\sigma_k')^2 + \sigma_k^2\right).$$

Therefore, the proposal distributions cancel in the Metropolis-Hastings acceptance probability. Again trial and error showed that $\sigma_\sigma = 50.0$ is a good choice because it yields smaller estimated integrated autocorrelation times $\hat{\tau}(\sigma_k^2)$, $k = 1, 2, 3$, than the ones achieved by $\sigma_\sigma = 40.0$ and $\sigma_\sigma = 60.0$. The weights were also updated by a Metropolis-Hastings algorithm. The proposed weights $\{w_k'\}_{k \in \underline{K}}$ were drawn jointly from the prior distribution

$$\{w_k'\}_{k \in \underline{K}} \sim \mathit{Dirichlet}(1, \ldots, 1)$$

independently of the current weights. For this proposal mechanism, no tuning is required. It remains to describe the order of updates. For reversibility, first a joint weight update was attempted, then component means and variances were updated in random order (six draws out of $\{\mu_1, \mu_2, \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2\}$ with replacement), and finally another joint update (w_1, w_2, w_3) for the weights was carried out. The algorithm was run for $N = 100\,000$ iterations (not including the burn-in of 10 000 iterations). To see the difference in mixing between the hottest distribution and the target distribution, the same MCMC algorithm was run under the target temperature. As discussed earlier, when sampling from the hottest distribution, labels are switched frequently so that the histograms for the weights, the means and the variances are symmetric in all components, respectively (see Figures 7-3, 7-5 and 7-7). At the target temperature, virtually no label switching takes place so that the components can be clearly distinguished (see Figures 7-4, 7-6 and 7-8). Due to the lack of label switching, the window estimator for the integrated autocorrelation time did not converge.

7.7 Approximating the curve

7.7.1 Constructing a decreasing interpolation

Having found an appropriate hottest temperature $\beta_{\min} = \frac{1}{8}$, we can now move on to finding an optimal temperature scheme $\beta_{\min} < \beta_n < \ldots < \beta_1 = \beta_0$ in tempered transitions. For this, we need a way of approximating the curve $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ because this curve is unknown due to the complexity of the mixture model. We will discuss interpolating $g(\beta)$ in two sections. In this section, we will assume that the exact values of the curve and its derivative are known at some anchor points between the hottest and the coldest temperature and then explain how to fit an interpolating curve between these anchor points. In the following section, we will discuss how we can estimate the required anchor points when these are not analytically or numerically available.

Let us start with assuming that the exact values of the curve $g(\beta)$ and its derivative $g'(\beta)$ are available at the anchor points $\beta_{\min} = \tilde{\beta}_m < \cdots < \tilde{\beta}_1 = \beta_0$. We mark these anchor points deliberately by a tilde to distinguish them from the inverse temperatures used in tempered transitions. Similarly, the number of anchor points m differs from the number n of temperatures employed in tempered transitions. We can use these anchor points to construct an interpolating curve $\hat{g}(\beta)$ between $\beta_{\min}$ and β_0. We know from theory that the curve $g(\beta)$ is decreasing (Section 5.3.1) so that we require the interpolating curve $\hat{g}(\beta)$ to be decreasing. We will interpolate the curve piecewise between adjacent anchor points by cubic Hermite interpolation because this interpolation makes use of all the available information at these points. Unfortunately, cubic interpolation does not guarantee a decreasing interpolation; the curve may contain a local optimum between two anchor points, but that can easily be checked. To ensure the resulting approximation is decreasing, an "emergency" solution may be to replace the non-decreasing part of the cubic interpolation automatically by linear interpolation. A better way is to adjust the spacing such that the anchor points are more densely spaced where the curve is strongly decaying. This could be achieved either by changing the spacing between anchor points as appropriate (e.g. from equidistant to geometric) or by placing more anchor points in the problematic patch.

We will now describe the cubic Hermite interpolation following the general

algorithm for polynomial Hermite interpolation (Burden and Faires 2001, pp. 137–139). Given $\hat{g}(\tilde{\beta}_i)$ and $\hat{g}'(\tilde{\beta}_i)$ for $i = 1, \ldots, m$ (where $\tilde{\beta}_i < \tilde{\beta}_{i-1}$), we will obtain a piecewise cubic Hermite interpolation by fitting

$$\hat{g}(\beta) = Q_{0,0} + Q_{1,1}\,(\beta - \tilde{\beta}_i) + Q_{2,2}\,(\beta - \tilde{\beta}_i)^2 + Q_{3,3}\,(\beta - \tilde{\beta}_i)^2\,(\beta - \tilde{\beta}_{i-1}) \tag{7.3}$$

on each patch $[\tilde{\beta}_i, \tilde{\beta}_{i-1}]$, which is based on four bits of information, namely on $\hat{g}(\tilde{\beta}_i)$, $\hat{g}(\tilde{\beta}_{i-1})$, $\hat{g}'(\tilde{\beta}_i)$ and $\hat{g}'(\tilde{\beta}_{i-1})$. The $Q_{j,j}$'s are defined recursively. We can only determine them by calculating all of the following quantities:

$$\begin{aligned}
Q_{2j,0} &= \hat{g}(\tilde{\beta}_{i-j}), && j = 0, 1,\\
Q_{1,0} &= \hat{g}(\tilde{\beta}_i),\\
Q_{2j+1,1} &= \hat{g}'(\tilde{\beta}_{i-j}), && j = 0, 1,\\
Q_{2,1} &= \frac{Q_{2,0} - Q_{1,0}}{\tilde{\beta}_{i-1} - \tilde{\beta}_i},\\
Q_{j,k} &= \frac{Q_{j,k-1} - Q_{j-1,k-1}}{\tilde{\beta}_{i-1} - \tilde{\beta}_i}, && (j,k) = (2,2), (3,2), (3,3).
\end{aligned}$$

The Hermite interpolation (7.3) is equivalent to the standard polynomial form

$$\hat{g}(\beta) = d_i\beta^3 + c_i\beta^2 + b_i\beta + a_i, \qquad \beta \in [\tilde{\beta}_i, \tilde{\beta}_{i-1}],$$

where

$$\begin{aligned}
a_i &= Q_{0,0} - Q_{1,1}\,\tilde{\beta}_i + Q_{2,2}\,\tilde{\beta}_i^2 - Q_{3,3}\,\tilde{\beta}_i^2\,\tilde{\beta}_{i-1},\\
b_i &= Q_{1,1} - 2\,Q_{2,2}\,\tilde{\beta}_i + Q_{3,3}\,\tilde{\beta}_i^2 + 2\,Q_{3,3}\,\tilde{\beta}_i\,\tilde{\beta}_{i-1},\\
c_i &= Q_{2,2} - 2\,Q_{3,3}\,\tilde{\beta}_i - Q_{3,3}\,\tilde{\beta}_{i-1},\\
d_i &= Q_{3,3}.
\end{aligned}$$

The polynomial form contains more terms and is therefore more expensive than the previous form (7.3), but it helps here to find the local optima

$$\beta_{\mathrm{opt}_1,\,\mathrm{opt}_2} = \frac{-c_i \pm \sqrt{c_i^2 - 3\,b_i\,d_i}}{3\,d_i}$$

of $\hat{g}(\beta) = d_i\beta^3 + c_i\beta^2 + b_i\beta + a_i$ if the optima exist. If $\beta_{\mathrm{opt}_1} \in (\tilde{\beta}_i, \tilde{\beta}_{i-1})$ or $\beta_{\mathrm{opt}_2} \in (\tilde{\beta}_i, \tilde{\beta}_{i-1})$, then we will replace the fitted $\hat{g}(\beta)$ for this particular interval $[\tilde{\beta}_i, \tilde{\beta}_{i-1}]$ by the linear interpolation

$$\hat{g}(\beta) = b_i\,\beta + a_i, \qquad \beta \in [\tilde{\beta}_i, \tilde{\beta}_{i-1}].$$

The parameters a_i and b_i are the unique solutions of the linear equation system

$$\begin{aligned}
\hat{g}(\tilde{\beta}_i) &= b_i\,\tilde{\beta}_i + a_i\\
\hat{g}(\tilde{\beta}_{i-1}) &= b_i\,\tilde{\beta}_{i-1} + a_i,
\end{aligned}$$

namely

$$\begin{aligned}
b_i &= \frac{\hat{g}(\tilde{\beta}_{i-1}) - \hat{g}(\tilde{\beta}_i)}{\tilde{\beta}_{i-1} - \tilde{\beta}_i}\\
a_i &= \hat{g}(\tilde{\beta}_i) - b_i\tilde{\beta}_i.
\end{aligned}$$

To unify notation, the linear interpolation will then be written by

$$\hat{g}(\beta) = d_i\,\beta^3 + c_i\,\beta^2 + b_i\,\beta + a_i, \qquad \beta \in [\tilde{\beta}_i, \tilde{\beta}_{i-1}],$$

where a_i and b_i are the parameters of the linear interpolation and the other parameters $c_i = 0$ and $d_i = 0$ are set to zero. As a result, we will have a fitted $\hat{g}(\beta)$ on $[\tilde{\beta}_m, \tilde{\beta}_1]$ that consists of cubic or linear, but always decreasing approximations on the patches $[\tilde{\beta}_i, \tilde{\beta}_{i-1}]$, $i = m, m-1, \ldots, 2$. This cubic/linear interpolation can then be represented by the $((m-1) \times 6)$ matrix

$$\begin{pmatrix} \tilde{\beta}_m & \tilde{\beta}_{m-1} & a_m & b_m & c_m & d_m \\ \tilde{\beta}_{m-1} & \tilde{\beta}_{m-2} & a_{m-1} & b_{m-1} & c_{m-1} & d_{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{\beta}_i & \tilde{\beta}_{i-1} & a_i & b_i & c_i & d_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \tilde{\beta}_2 & \tilde{\beta}_1 & a_1 & b_1 & c_1 & d_1 \end{pmatrix}.$$

To evaluate $\hat{g}(\beta)$ for any $\beta \in [\tilde{\beta}_m, \tilde{\beta}_0]$, we will first find the patch containing β. We can determine j such that $\beta \in [\tilde{\beta}_j, \tilde{\beta}_{j-1}]$ by the following algorithm:

Algorithm 7.1:

$j = 1$
$\mathsf{while}(\beta \leq \tilde{\beta}_j)\ \{\ j = j + 1\ \}$
$\mathsf{return}(j)$

If we know the relevant patch j, it is straightforward to evaluate the desired approximation $\hat{g}(\beta) = d_j\beta^3 + c_j\beta^2 + b_j\beta + a_j$.

For illustration, the curve $g(\beta)$ will be interpolated in a case where the curve $g(\beta)$ and its derivative $g'(\beta)$ can be calculated analytically. This has two advantages: first, we can check the quality of interpolation; and second, we can ignore the problem of estimating the anchor point values for the time being. Let us return to the earlier example of tempering a d-dimensional multivariate standard normal distribution discussed in Section 5.4.1. We derived there the expectation and variance of the energy function $h(x) = \frac{1}{2}\sum_{j=0}^{d} y_j^2$ under the tempered distribution. They are $\mathbb{E}_{p_\beta}[h(X)] = \frac{d}{2\,\beta}$ and $\mathrm{var}_{p_\beta}[h(X)] = \frac{d}{2\,\beta^2}$. For $d = 1$, this implies

$$\begin{aligned} g(\beta) &= \mathbb{E}_{p_\beta}[h(X)] \\ &= \frac{1}{2\,\beta} \\ g'(\beta) &= -\mathrm{var}_{p_\beta}[h(X)] \\ &= -\frac{1}{2\,\beta^2}, \end{aligned}$$

which is all we need for testing the interpolation method.

The task is now to interpolate $g(\beta) = \frac{1}{2\beta}$ between $\left[\frac{1}{16}, 1\right]$ based on few anchor points. In a first attempt, a solely cubic interpolation is chosen. It is based on five equally spaced anchor points in $\left[\frac{1}{16}, 1\right]$. Unfortunately, the resulting approximation does not satisfy the constraint of being decreasing because it features a local optimum (see Figure 7-9 top left). Replacing the unsatisfactory patch by a linear interpolation yields a decreasing approximation; however, the fit of the linear piece is relatively bad (see Figure 7-9 top right). Satisfactory interpolations are obtained if the anchor points support the interpolation in the problematic area better. This can either be achieved by adding an additional anchor point in the problematic area (see Figure 7-9 bottom left) or by a more appropriate spacing (see Figure 7-9 bottom right), which is here geometric. If the support is better, no linear replacements are required so that the interpolation is purely piecewise cubic. As a general rule, we could start with spacing anchor points geometrically by default. Such a spacing would be ideal to interpolate a convex curve as more anchor points are placed where the curve strongly decays. We can then check whether the spacing is good by plotting the anchor points. If the plot suggests a different shape of the curve, we may change or add anchor points accordingly. Once the final spacing is chosen, we can interpolate the curve. If the fit does not look smooth, we can add anchor points where required and repeat the interpolation.

7.7.2 Obtaining anchor points

If we are not able to obtain the anchor points $\hat{g}(\tilde{\beta}_i)$ and their derivatives $\hat{g}'(\tilde{\beta}_i)$, $i = 1, \ldots, m$, for the interpolation analytically or numerically, we may estimate them by importance sampling. As suggested by Jennison (1993), we can use a tempered version of the target distribution as importance sampling distribution. For this estimation, we have different target distributions p_{β_i}, $i = 1, \ldots, m-1$, all of which are covered by the hottest distribution $p_{\beta_{\min}}$. Recall that importance sampling only requires a single large sample from the importance distribution because the same importance sample can be used to estimate the expectations of different functions under the same target distribution as well as under different target distributions. For convenience, we will therefore take a single large sample from $p_{\beta_{\min}}$ to estimate the anchor points $\hat{g}(\tilde{\beta}_i)$ and their derivatives $\hat{g}'(\tilde{\beta}_i)$ for all $i = 1, \ldots, m-1$. This sample will also be used to estimate the last point $\hat{g}(\tilde{\beta}_m)$ and its derivative $\hat{g}'(\tilde{\beta}_m)$

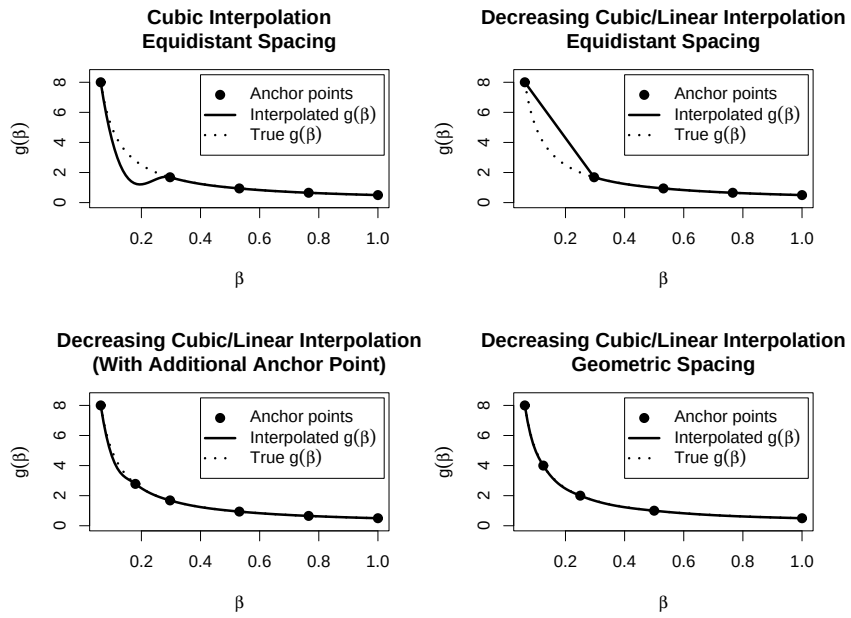

Figure 7-9: The purely piecewise cubic interpolation *(top left)* may lead to undesired local optima. The relevant patch may then be replaced by a linear interpolation *(top right)*. However, additional anchor points *(bottom left)* or appropriately spaced anchor points *(bottom right)* offer more satisfactory solutions.

by simple MCMC estimation, which is here appropriate because, at the last anchor point $\tilde{\beta}_m = \beta_{\min}$, the target distribution and the importance sampling distribution are identical.

The main concern with importance sampling is that the importance sampling distribution does not cover the target distribution closely enough. In that case, most of the importance samples cover the tails of the target distribution and have therefore tiny weights. The few importance samples that cover the modes of the target distribution will have much larger weights and thus a great impact on the importance estimate (Robert and Casella 1999, Section 3.3). Large weights lead to jumps in the trace plots of the importance estimate so that the convergence of the importance estimate to the theoretical expectation will be slower. When using the hottest distribution as importance sampling distribution to estimate $\hat{g}(\tilde{\beta}_i)$ and $\hat{g}'(\tilde{\beta}_i)$ for $\beta_{\min} = \tilde{\beta}_m < \cdots < \tilde{\beta}_1 = \beta_0$, the greatest divergence between the importance sampling distribution and the target distribution occurs when estimating $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ at $\tilde{\beta}_1 = 1$ so that this case gives the worst accuracy of importance sampling. To get a feeling for the accuracy, let us investigate the worst case scenario of

estimating $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ in an example where the true curve $g(\beta)$ and thus the true values $g(\tilde{\beta}_1)$ and its derivative $g'(\tilde{\beta}_1)$ are known. We will again consider the earlier toy example of tempering a standard normal distribution in which the tempered distribution is also normal, but with variance $\frac{1}{\beta}$, i.e. $p_\beta(x) \sim N\left(0, \frac{1}{\beta}\right)$. As derived in Section 5.4.1, the true curve is $g(\beta) = \frac{1}{2\beta}$ and its derivative is $g'(\beta) = \frac{1}{2\beta^2}$. As this example is quite well behaved for estimating $g(\tilde{\beta}_1) = \mathbb{E}_{p_{\tilde{\beta}_1}}[h(X)]$ and $g(\tilde{\beta}_1) = -\mathrm{var}_{p_{\tilde{\beta}_1}}[h(X)]$ by importance sampling based on a sample from $p_{\beta_{\min}}(x)$, we have to choose $\beta_{\min}$ very small (here $\beta_{\min} = 10^{-9}$) to obtain extreme importance weights. Plotting the importance weights as well as the trace plots of the importance estimates $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ shows that extremely large weights cause jumps in the importance estimates (see Figure 7-10). Here only 1% of the weights are positive, the others are computationally equal to zero, so that the importance estimates are based on 1% of the samples, but averaged over the total number N of iterations (here $N = 100\,000$). Despite the jumps, the method converges. The last values are here $\hat{g}(\tilde{\beta}_1) = 0.4$ and $\hat{g}'(\tilde{\beta}_1) = 0.41$. Bearing in mind that the importance sampling distribution $N(0, 10^9)$ is extremely flat, these estimates are relatively good approximations to the true values $g(\tilde{\beta}_1) = 0.5$ and $g'(\tilde{\beta}_1) = 0.5$.

For comparison, let us also investigate the worst fit of the importance sampling distribution for our galaxy model. The divergence between the tempered versions

$$p_\beta\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}} \middle| \{y_j\}_{j\in\underline{n}}\right) \propto p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\left[p\left(\{y_j\}_{j\in\underline{n}} \middle| \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\right]^\beta$$

is largest between the hottest distribution $p_{\beta_{\min}}$ (here at $\beta_{\min} = \frac{1}{8}$) and the target distribution $p_{\tilde{\beta}_1}$ (at $\tilde{\beta}_1 = 1$). Again trace plots of the importance weights and importance estimates $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ reveal that jumps are caused by relatively large weights (see Figure 7-11). However, both estimates seem to settle within the given $N = 100\,000$ iterations. The last values are $\hat{g}(\tilde{\beta}_1) = 209.5$ and $\hat{g}'(\tilde{\beta}_1) = -4.6$. Later we will see that the corresponding MCMC estimates obtained by tempered transitions lie by $\hat{g}(\tilde{\beta}_1) = 208.6$ and $\hat{g}'(\tilde{\beta}_1) = -5.2$. In that light, the importance estimates seem quite usable first approximations.

A nice feature of the energy function $h(\cdot)$ is that we can deduce the importance weights from it. To estimate the mean and the variance of the energy, we only need the output chain $\left\{h\left(X^{(t)}\right)\right\}_{t=1}^N$ when sampling from the hottest

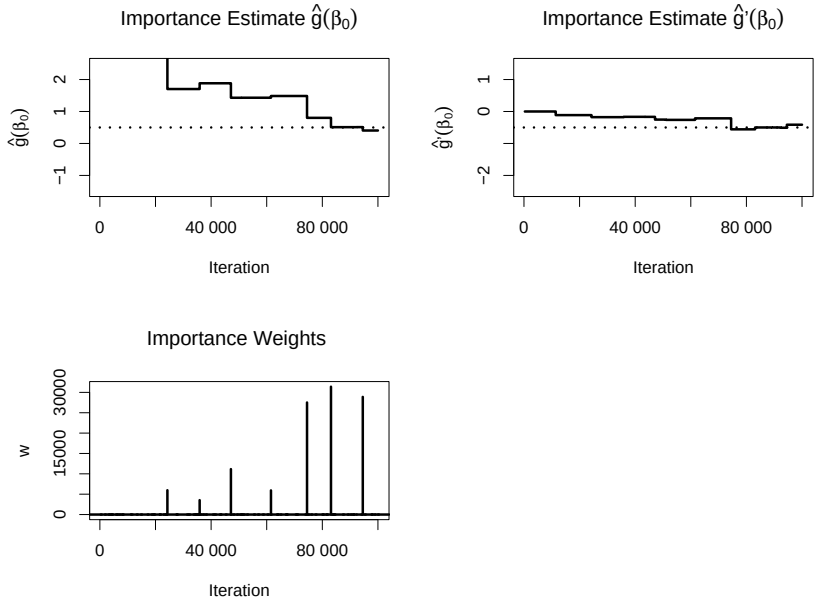

Figure 7-10: The running importance estimates $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ and the importance weights based on a flat importance sampling distribution (toy example). Despite the jumps caused by the extreme weights, the estimates *(solid lines)* converge to the true values *(dotted lines)*.

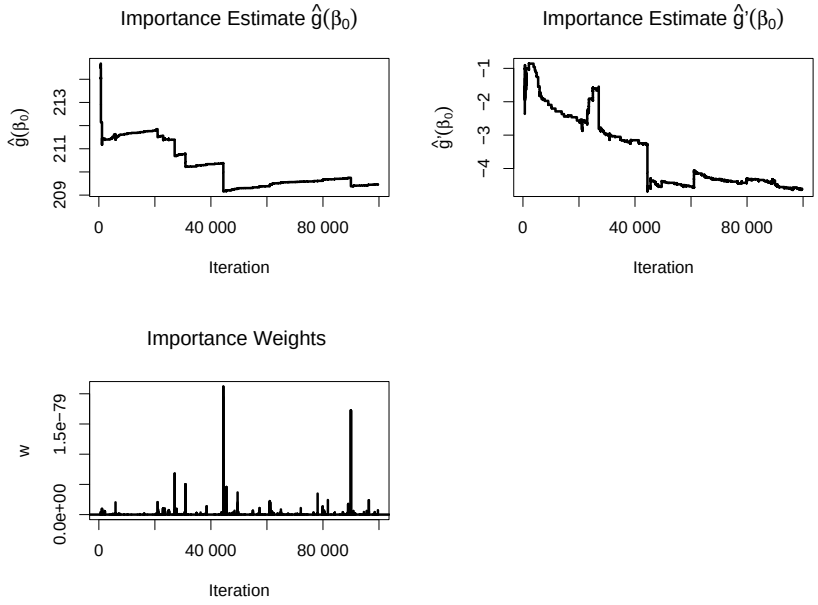

Figure 7-11: The running importance estimates $\hat{g}(\tilde{\beta}_1)$ and $\hat{g}'(\tilde{\beta}_1)$ and the importance weights in the "galaxy" example where the estimation is based on samples from the hottest distribution. Despite the jumps caused by the extreme weights, the estimates converge.

distribution. From this chain, the importance weights

$$\begin{aligned} w_t &= \frac{p_\beta\left(X^{(t)}\right)}{p_{\beta_{\min}}\left(X^{(t)}\right)} \\ &= \frac{\pi\left(X^{(t)}\right)\exp\left[-\beta\, h\left(X^{(t)}\right)\right]}{\pi\left(X^{(t)}\right)\exp\left[-\beta_{\min}\, h\left(X^{(t)}\right)\right]} \\ &= \exp\left[-\left(\beta-\beta_{\min}\right) h\left(X^{(t)}\right)\right] \end{aligned}$$

can be determined due to the special tempering structure. Since the energy chain takes univariate real values in any application, the code has to be written only once. After that, it can be used for any problem.

Let us now move on to examining the quality and variability of the interpolation and how far these affect the resulting optimal temperature scheme.

7.8 Testing the robustness of interpolation

7.8.1 Key issues

There are two key issues connected to the importance-sampling-based interpolation: the accuracy of interpolation and the variability of the optimisation results that are based on the approximated curve. To test the accuracy, we will check the variability of interpolation as well as the quality of prediction. Since the curve $g(\beta)$ is unknown, we cannot test the interpolation against the true values. We can however estimate them based on a sample generated by tempered transitions. We will therefore start with specifying the tempered transitions algorithm in Section 7.8.2 before moving on to testing the quality of interpolation in Section 7.8.3 and its effect on the optimisation in Section 7.8.4.

7.8.2 Tempered transitions set-up

To check the accuracy of the interpolation, we want to see how well $g(\beta)$ and $g'(\beta)$ are predicted between the anchor points of the interpolation. As $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ and $g'(\beta) = -\mathrm{var}_{p_\beta}[h(X)]$ are not analytically available, we will approximate these values for some β values by MCMC estimation based on samples from the corresponding tempered posterior distribution p_β. Since the tempered posterior distributions are multimodal, we will apply tempered transitions. As the interpolated curves are here convex (see Figure 7-12), we will use a geometric scheme $\beta_{\min} = \beta_n < \beta_{n-1} < \cdots < \beta_2 < \beta_1 = \beta_0$. Finding

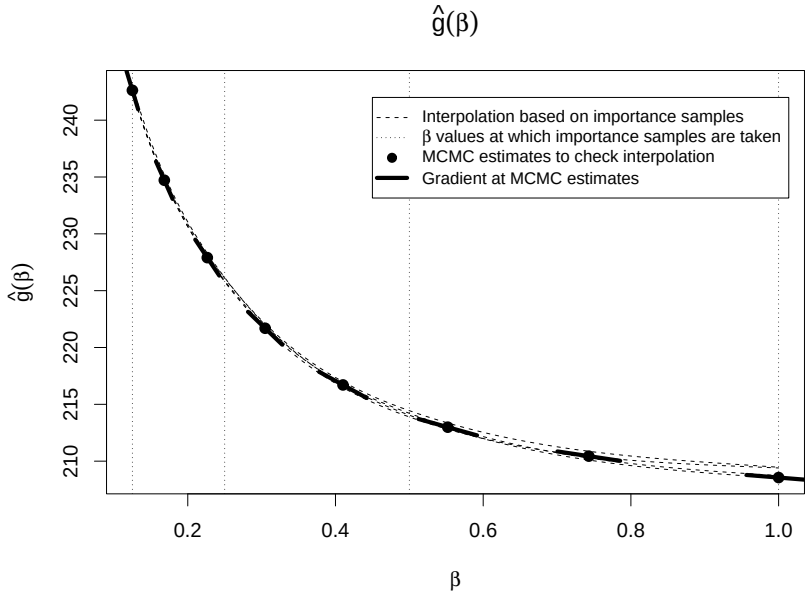

Figure 7-12: The interpolation of $g(\beta)$ is based on the importance estimates $\hat{g}(\tilde{\beta}_i)$ and $\hat{g}'(\tilde{\beta}_i)$, taken at $\tilde{\beta}_i = 8^{-(i-1)/3}$, $i = 1, 2, 3, 4$, *(vertical dotted lines)*. The estimates are based on the same sample from the hottest distribution $p_{\beta_{\min}}$. The four replicate interpolations *(dotted curves)*, each based on a different sample from $p_{\beta_{\min}}$, show little variation. The interpolations predict values in-between the anchor points quite well as a comparison with more accurate MCMC estimates $\hat{g}(\tilde{\beta}_i)$ *(dots)* and $\hat{g}'(\tilde{\beta}_i)$ *(gradients at dots)* at $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, \ldots, 8$, reveals.

an optimal scheme is not attempted for this would involve the interpolation whose quality is under investigation. Note that we double again the first temperature by setting $\beta_1 = \beta_0$ for ease of presentation as explained in Section 4.2. Further note that this time the target temperature varies since now various tempered versions of the posterior are the target distributions. Suppose we want to estimate $g(\beta)$ and $g'(\beta)$ at $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, \ldots, 7$, then we will employ tempered transitions for each $\tilde{\beta}_i$ separately by setting the target temperature equal to the respective temperature $\beta_0 = \tilde{\beta}_i$. We have already chosen the hottest temperature $\beta_{\min} = \frac{1}{8}$ and a corresponding standard MCMC kernel in Section 7.6. We will use similar standard MCMC kernels when generating the auxiliary states under the various tempered distributions in the proposal mechanism of tempered transitions. That means, at each temperature β_i, we will update the component weights jointly by a Metropolis-Hastings

kernel with independent proposal from the Dirichlet prior

$$\{w'_k\}_{k\in\underline{K}} \sim Dirichlet(1,\ldots,1).$$

This kernel does not change with the temperature. The component means will be updated one by one by a Metropolis kernel based on normally distributed proposals

$$\mu'_k \sim N\left(\mu_k, \left(\sigma_\mu^{(i)}\right)^2\right)$$

where the step size $\sigma_\mu^{(i)}$ may vary with the temperature β_i. The step size at the hottest temperature will be $\sigma_\mu^{(n)} = 25.0$ as before. We will try different step size plans below. Similarly, at each temperature level β_i, new component variances will be chosen one by one by a Metropolis-Hastings kernel with reflected draw from a normal distribution in which any negative values are made positive by the following mechanism: if σ_k^2 is the current state, a proposal state $(\sigma'_k)^2$ is created by first drawing

$$z \sim N\left(0, \left(\sigma_\sigma^{(i)}\right)^2\right)$$

and then setting

$$(\sigma'_k)^2 = \begin{cases} \sigma_k^2 + z & \text{if } \sigma_k^2 + z \geq 0 \\ -(\sigma_k^2 + z) & \text{otherwise.} \end{cases}$$

As the corresponding proposal distribution is symmetric, it cancels in the Metropolis-Hastings acceptance ratio. The step size $\left(\sigma_\sigma^{(i)}\right)$ may depend on the temperature level β_i with $\left(\sigma_\sigma^{(n)}\right) = 50.0$ as before. For reversibility, updates are carried out in the following order at each temperature level: first new weights are proposed jointly, then the means and variances are updated in random order (six draws out of $\{\mu_1, \mu_2, \mu_3, \sigma_1^2, \sigma_2^2, \sigma_3^2\}$ with replacement), followed by another joint update for the weights.

Now we need to find a reasonable step size plan. We will try three plans, namely constant step sizes, step sizes which increase independently of the temperature scheme and step sizes which increase in line with the temperature scheme. The constant step size plan

$$\sigma^{(i)} = \sigma^{(n)} \qquad i = 1,\ldots,n,$$

keeps the same large step size $\sigma^{(n)}$ at all temperature levels in the hope that the tempered transitions sampler travels far. The second and the third plan both

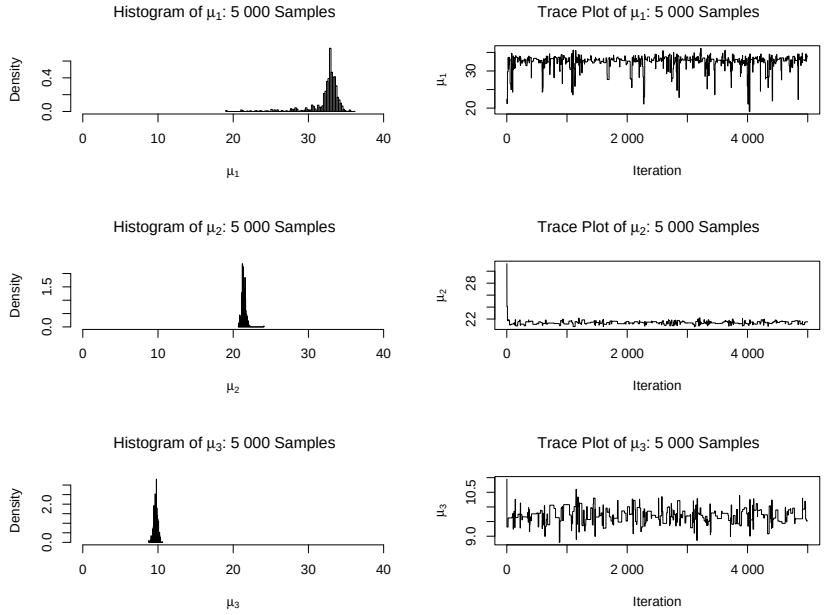

Figure 7-13: Brief preliminary tempered transitions run with constant step size plan. The algorithm does not mix well between labels.

increase the step size from small step sizes at cold temperatures to large step sizes at hot temperatures in the hope that the sampler takes a large step to a new mode at the hottest temperature and then takes slowly smaller getting steps to explore this new mode in the cooling down process. The increase of the second step size plan

$$\left(\sigma^{(i)}\right)^2 = \left(\sigma^{(1)}\right)^2 + \frac{\left(\sigma^{(n)}\right)^2 - \left(\sigma^{(1)}\right)^2}{n}\, i, \qquad i = 1, \ldots, n,$$

is such that the squared step sizes $\left(\sigma^{(i)}\right)^2$ increase linearly with the number of temperature levels, but independently of the value of the temperature levels. This plan performed well in the Witch's Hat toy example (Section 6.5) and is therefore chosen. The third plan

$$\sigma^{(i)} = \sigma^{(1)} + \frac{\sigma^{(n)} - \sigma^{(1)}}{\sqrt{1/\beta_n} - \sqrt{1/\beta_1}} \left(\sqrt{1/\beta_i} - \sqrt{1/\beta_1}\right), \qquad i = 1, \ldots, n,$$

adapts the step sizes dependent on the temperature levels. It mirrors the way in which the standard deviation of a normal distribution grows when the distribution is heated. This is a sensible approach here because the proposal mechanisms for updating the component means and variances involve drawing from normal distributions centred at the current value. In other words, the third plan slowly shrinks the MCMC kernel for the hottest distribution to fit the colder distributions. In all plans, the step sizes at the hottest temperature

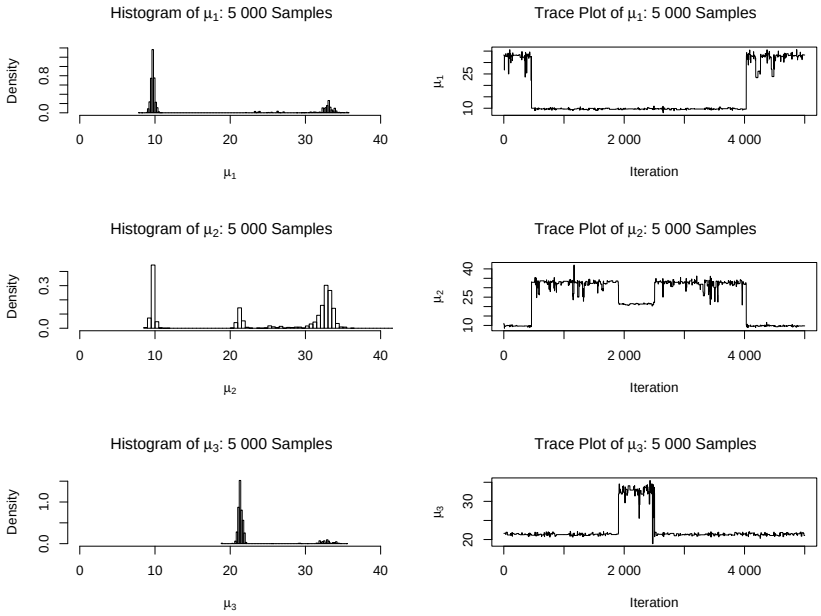

Figure 7-14: Brief preliminary tempered transitions run with a temperature-independent increase in the step size plan. The algorithm mixes between labels.

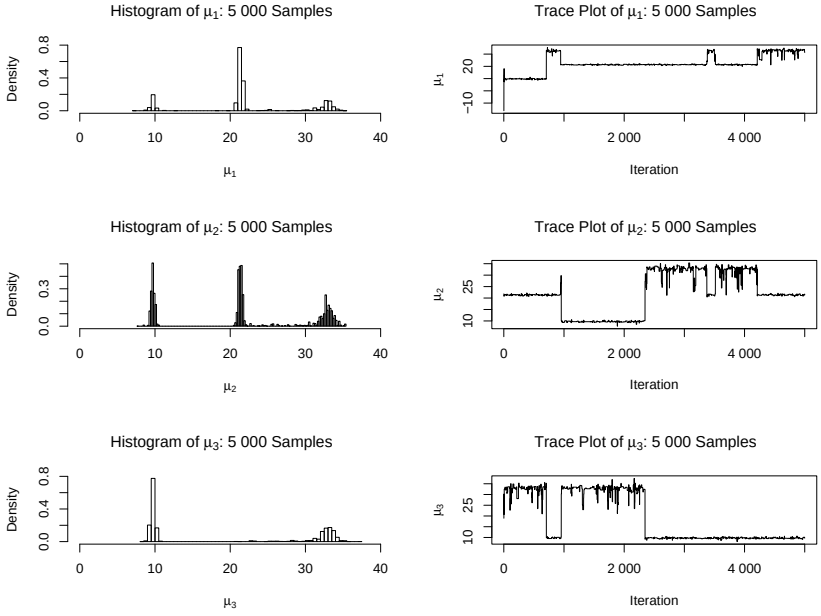

Figure 7-15: Brief preliminary tempered transitions run with an adaptive increase in the step size plan. The algorithm mixes between labels. Compared to the temperature-independent plan (Figure 7-14), the mixing is here slightly faster.

are set to the previously determined $\sigma_\mu^{(n)} = 25.0$ and $\sigma_{\sigma^2}^{(n)} = 50.0$. For the second and the third plan, the step sizes at the target temperature also have to be chosen. Looking at the histograms of the standard MCMC sample at target temperature (see Figures 7-6 and 7-8), it seems that $\sigma_\mu^{(1)} = 1.0$ and $\sigma_{\sigma^2}^{(1)} = 2.0$ are reasonable choices to allow local exploration of modes. Note that this choice also maintains the ratio of σ_μ and σ_{σ^2}: at each temperature, the step size σ_{σ^2} is twice as big as σ_μ.

The three step plans will be tested on sampling from the coldest distribution as this is the hardest sampling problem. As a first guess, the number of geometric temperatures is set to $n = 20$. Since this already defines an expensive algorithm, the step size plans will be tested on brief preliminary runs of 5 000 iterations. This is not enough for convergence so we cannot use the integrated autocorrelation time as an efficiency measure. We will instead assess the quality of mixing graphically by histograms and trace plots: in this brief run, the first plan (constant large step size) does not switch labels at all, while the second plan (linear increase of the step size variance) and the third plan (temperature-adapted increase of the step size) mix between labels (see Figures 7-13 to 7-15). As the mode swapping seems best under the third plan, we will choose this for the long runs of tempered transitions when sampling from the tempered distribution p_β at various β values later. Note that all three plans give similar acceptance rates of approximately 0.14.

7.8.3 Quality of interpolation

To get a feeling for the variability of interpolation, we will repeat the interpolation process four times and plot the resulting curves. At the beginning of each interpolation, a sample of size $N = 100\,000$ is generated from the hottest distribution by standard MCMC as discussed in Section 7.6. The same sample is used to estimate the anchor points $\hat{g}(\tilde{\beta}_i)$ and their derivatives $\hat{g}'(\tilde{\beta}_i)$ at $\tilde{\beta}_i = 8^{-(i-1)/3}$, $i = 1, 2, 3, 4$. Then, the curve $g(\beta)$ is approximated piecewise between these anchor points by cubic Hermite interpolation as discussed in Section 7.7.1. Any variation between these four replicate interpolations is thus solely caused by the fact that each interpolation comes from a different sample from the hottest distribution. Plotting the four replicate curves shows that the curves lie close together at temperatures close to the hottest temperatures, but then spread out slightly as β approaches the coldest temperature $\beta_0 = 1$ at which the variability is greatest (see Figure 7-12 displayed earlier). This

behaviour can be explained by importance sampling becoming less accurate the further the distance between the importance sampling distribution (here the hottest distribution) and the target distribution (here the tempered distribution at temperature $\tilde{\beta}_i = 8^{-(i-1)/3}$, $i = 1, 2, 3$). The first impression is that the slight variability does not matter much since all the curves take very similar courses. To verify that this variability is of no importance, we will compare the sets of optimal temperatures obtained when the optimisation is based on the different interpolated curves in Section 7.8.4.

Another way of assessing the quality of interpolation is to test how well the interpolation predicts points between the anchor points. We will check the points $\hat{g}(\tilde{\beta}_i)$ and their derivatives $\hat{g}'(\tilde{\beta}_i)$ at $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, 2, 3, 4, 5, 6, 7, 8$. Since these points are not analytically available, we will estimate them by MCMC estimation. At each point, again four replicates were taken to assess the variability of the MCMC estimates. The MCMC estimates at $\tilde{\beta}_8 = \frac{1}{8}$ are based on samples obtained by standard MCMC because $\tilde{\beta}_8$ is equal to the hottest temperature at which standard sampling is possible. To sample from the tempered distributions at temperatures $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, 2, 3, 4, 5, 6, 7$, tempered transitions is employed based on $n = 16, 16, 4, 4, 2, 2, 2$ geometrically spaced temperatures between $\beta_{\min} = \tilde{\beta}_m$ and $\beta_0 = \tilde{\beta}_i$ respectively. The acceptance rates are $0.13, 0.19, 0.43, 0.49, 0.62, 0.71, 0.85$ respectively. We will check how well the interpolation predicts the points $\hat{g}(\tilde{\beta}_i)$ and their derivatives $\hat{g}'(\tilde{\beta}_i)$ at $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, 2, 3, 4, 5, 6, 7, 8$, graphically. Originally it was thought to add all MCMC replicates to the plot. But this was impossible because the replicates (see Table 7-1) lie so close together that they could hardly be distinguished in the plot. Instead, the mean values of the replicates at each $\tilde{\beta}_i$ are displayed in Figure 7-12: the mean of the $\hat{g}\left(\tilde{\beta}_i\right)$ replicates is represented by a dot, while the mean of the replicate derivatives $\hat{g}'\left(\tilde{\beta}_i\right)$ is represented by a short tangent. The prediction is quite good; all the MCMC estimates lie within the narrow belt produced by the spread of the interpolated curves. We can infer from this that the interpolated curve imitates the true behaviour of the curve.

We have seen that the MCMC estimates of the anchor points are more accurate than the importance sampling estimates. One may therefore consider always interpolating the curve based on MCMC estimates rather than importance estimates. The problem is that the MCMC-based interpolation is very

i	$\hat{g}\left(\tilde{\beta}_i\right)$ (4 replicates)			
1	209.5	208.5	208.7	209.4
2	211.1	210.2	210.4	210.6
3	213.4	212.9	213.1	212.8
4	217.1	216.8	216.9	216.5
5	222.1	222.0	221.9	221.7
6	228.2	228.3	228.0	228.0
7	235.0	235.3	234.9	234.9
8	242.5	243.0	242.5	242.5

Table 7-1: The MCMC estimates $\hat{g}\left(\tilde{\beta}_i\right)$ at $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, \ldots, 8$, are based on tempered transitions samples. The replicates lie close together.

expensive since the MCMC samples are generated by tempered transitions. Let us compare the cost of the importance sampling approach with the one of the tempered transitions approach. Since the importance estimates are all based on the same importance sample, the cost of obtaining anchor points by importance samples consists here of one MCMC run and three applications of importance sampling. The three importance sampling applications take together less than the one MCMC run. For simplicity, let us assume that the three applications cost together as much as the MCMC run at the hottest distribution. The cost for the importance sampling approach corresponds thus to the cost of two simple MCMC runs. Now let us approximate the cost for the tempered-transitions-based interpolation approach. Here the tempered transitions runs to sample from the tempered distribution at $\tilde{\beta}_i$, $i = 1, 2, 3, 4, 5, 6, 7$, are based on $n = 16, 16, 4, 4, 2, 2, 2$ geometrically spaced temperatures. As tempered transitions with n temperatures is $2n$ times as expensive as a simple MCMC run, the cost of all these tempered transitions runs corresponds to the cost of 92 simple MCMC runs. We also have to add the cost of the simple MCMC run at the hottest distribution $\tilde{\beta}_8 = \frac{1}{8}$. It turns out that the MCMC-based approach is 93 times as expensive as a simple MCMC run or, equivalently, 46.5 times as expensive as the above importance sampling approach. One may argue that this higher computational cost is due to the higher number of anchor points. Although reducing the number of MCMC sample points to four, which is the number of importance sampling estimates, decreases the cost of the MCMC-based interpolation, the resulting cost is still very high. Suppose we used the MCMC anchor points at the positions $\tilde{\beta}_i$, $i = 1, 4, 6, 8$, which are

comparable to the positions taken in the importance sampling approach. Then we would have the cost of one simple MCMC run at the hottest temperature $\tilde{\beta}_8 = \frac{1}{8}$ plus the cost of the tempered transitions runs for sampling from the distributions at temperatures $\tilde{\beta}_i$, $i = 1, 4, 6$, based on $n = 16, 4, 2$ temperatures respectively. By similar calculations as above, the total cost of the reduced MCMC approach turns out to be 45 times as high as the cost of a simple MCMC run or equivalently 22.5 times as high as the cost of the importance sampling approach. In conclusion, the interpolation based on importance samples is very cost-efficient and effective.

7.8.4 Effect on the temperature optimisation

The last robustness test is to check whether the slight variability of the importance-sampling-based interpolation matters when determining the best schedule for the tempered transitions algorithm and the associated minimal sum of squares. Again we cannot compare the optimisation results with the ones obtained if the optimisation is based on the true curve $g(\beta)$ because the curve is unknown. We can however use an interpolation based on the more accurate MCMC estimates from the previous section. To obtain an even higher accuracy, we will construct the benchmark interpolation by using the pooled mean of the four replicates that exist for every of the estimates $\hat{g}\left(\tilde{\beta}_i\right)$ and $\hat{g}'\left(\tilde{\beta}_i\right)$ where $\tilde{\beta}_i = 8^{-(i-1)/7}$, $i = 1, 2, 3, 4, 5, 6, 7, 8$. We will thus obtain a set $\{\beta_i^{\text{true}}\}_{i=1}^n$ giving the benchmark sum $S\left(\{\beta_i^{\text{true}}\}\right)$. We can assess how close any other scheme $\left\{\hat{\beta}_i^{\text{approx}}\right\}_{i=1}^n$ approximates the benchmark one by comparing the sum of squares $S\left(\{\beta_i^{\text{approx}}\}\right)$ [based on the benchmark interpolation of $g(\beta)$] with $S\left(\{\beta_i^{\text{true}}\}\right)$. We will use the optimal results obtained from the importance-sampling-based interpolations as well as the geometric scheme for $\left\{\hat{\beta}_i^{\text{approx}}\right\}_{i=1}^n$ for the variability of the former is under investigation and the latter is the default alternative. To assess the variability, it is sufficient to choose only two importance sampling curves, namely the one that lies furthest off the benchmark interpolation as well as the curve that lies furthest away from this extreme interpolation because these curves represent the worst accuracies. The extreme curves can be identified with respect to the distance measure $\left\{\sum_j \left[g_1(\tilde{\beta}_j) - g_2(\tilde{\beta}_j)\right]^2\right\}^{1/2}$ where g_1 and g_2 are the different interpolations and $\left\{\tilde{\beta}_j\right\}$ are 501 equidistant inverse temperatures between $\beta_{\min}$ and β_0. The corresponding approximating sequences are determined by dynamic programming. For a broader insight, the number of inverse temperatures

between $\beta_n = \frac{1}{8}$ and $\beta_1 = 1$ is varied by $n = 5, 16, 60$. The mesh size of the search is set to be 10^{-4}, implying that the returned optimal temperatures are accurate up to four decimal points. That means that the search grid is defined by $\{\beta_i\}_{i=1}^{n-1} \in \{0.125 + 10^{-4}\, d : \quad d = 0, 1, \ldots, 8\,750\}^{n-1}$. To check the accuracy of the mesh, the sum of squares for $n = 5$ and $n = 16$ geometric inverse temperatures can be compared with the sum of squares obtained by rounding these geometric inverse temperatures up to four decimal places (see Tables 7-2 and 7-3). The grid is fine enough because the error induced by the discretisation was less than 1‰. When optimising $n = 60$ temperatures, however, a mesh of size 10^{-3} is used to preserve the comparability up to a certain decimal point because the finer mesh size of 10^{-4} exceeds the storage of the computer program. Again the fineness of the 10^{-3} grid can be tested by comparing the sum of squares obtained by $n = 60$ geometric inverse temperatures and by $n = 60$ rounded geometric inverse temperatures (rounded up to three decimal places). The error lies by 1% (see Table 7-4), so the mesh size is also in order. Satisfied with the accuracy of the dynamic programming algorithm, we can now move on the actual robustness test where we assess the accuracy of the geometric rule and, more importantly, the accuracy of the importance-sampling-based optimisation by comparing the approximating sums $S(\{\beta_i^{\text{approx}}\})$ (based on the benchmark interpolation) with the benchmark sum $S(\{\beta_i^{\text{true}}\})$. The results can be found in the Tables 7-2 to 7-4. All three choices of $n = 5, 16, 60$ have similar accuracy so that the quality of estimation does apparently not depend on the number of temperatures. In each case, all methods applied to obtain $\{\beta_i^{\text{approx}}\}$ perform similarly; all the sums $S(\{\beta_i^{\text{approx}}\})$ differ less than 1% from the benchmark sum $S(\{\beta_i^{\text{true}}\})$. Within this slight variability, the geometric scheme is the worst approximation. The results show that the discretisation and the variability of the importance sampling interpolation are of no importance and also that the geometric schedule is close to optimal in this example. As this optimality refers to the related problem and not to the true efficiency problem, it remains to investigate how both schemes perform in the final tempered transitions run.

7.9 Final tempered transitions run

In the final run for sampling from the galaxy model, we will apply tempered transitions based on $n = 60$ between $\beta_n = \frac{1}{8}$ and $\beta_0 = 1$. In one version, the temperatures are set optimally (based on the benchmark interpolation); in

$\{\beta_i^{\text{approx}}\}$ obtained by	Sum of Squares for Various Sets of 5 Temperatures $S(\{\beta_i^{\text{approx}}\})$ based on MCMC interpolation
MCMC interpolation	5.72733
furthest off MCMC interpolation	5.72907
furthest off furthest interpolation	5.72773
geometric spacing	5.75643
rounded geometric spacing	5.75648

Table 7-2: The various sets $\{\beta_i^{\text{approx}}\}$ were obtained by optimising 5 inverse temperatures between $\beta_5 = \frac{1}{8}$ and $\beta_1 = 1$ on a mesh of size 10^{-4} with respect to (a) the benchmark MCMC interpolation, (b) the importance sampling interpolation which lies furthest off the benchmark interpolation, and (c) the importance sampling interpolation furthest off the furthest importance sampling interpolation. The remaining sets $\{\beta_i^{\text{approx}}\}$ were obtained by (d) geometric spacing and (e) rounding the geometrically spaced inverse temperatures up to four decimal places. Comparing the sum of squares with the benchmark sum of squares $S = 5.72733$ shows that all these sets are close to optimal in this example.

$\{\beta_i^{\text{approx}}\}$ obtained by	Sum of Squares for Various Sets of 16 Temperatures $S(\{\beta_i^{\text{approx}}\})$ based on MCMC interpolation
MCMC interpolation	1.50087
furthest off MCMC interpolation	1.50475
furthest off furthest interpolation	1.50346
geometric spacing	1.50885
rounded geometric spacing	1.50886

Table 7-3: The various sets $\{\beta_i^{\text{approx}}\}$ were obtained by optimising 16 inverse temperatures between $\beta_{16} = \frac{1}{8}$ and $\beta_1 = 1$ on a mesh of size 10^{-4} with respect to (a) the benchmark MCMC interpolation, (b) the importance sampling interpolation which lies furthest off the benchmark interpolation, and (c) the importance sampling interpolation furthest off the furthest importance sampling interpolation. The remaining sets $\{\beta_i^{\text{approx}}\}$ were obtained by (d) geometric spacing and (e) rounding the geometrically spaced inverse temperatures up to four decimal places. Comparing the sum of squares with the benchmark sum of squares $S = 1.50087$ shows that all these sets are close to optimal in this example.

$\{\beta_i^{\text{approx}}\}$ obtained by	Sum of Squares for Various Sets of 60 Temperatures $S(\{\beta_i^{\text{approx}}\})$ based on MCMC interpolation
MCMC interpolation	0.38142
furthest off MCMC interpolation	0.38260
furthest off furthest interpolation	0.38210
geometric spacing	0.38313
rounded geometric spacing	0.38354

Table 7-4: The various sets $\{\beta_i^{\text{approx}}\}$ were obtained by optimising 60 inverse temperatures between $\beta_{60} = \frac{1}{8}$ and $\beta_1 = 1$ on a mesh of size 10^{-3} with respect to (a) the benchmark MCMC interpolation, (b) the importance sampling interpolation which lies furthest off the benchmark interpolation, and (c) the importance sampling interpolation furthest off the furthest importance sampling interpolation. The remaining sets $\{\beta_i^{\text{approx}}\}$ were obtained by (d) geometric spacing and (e) rounding the geometrically spaced inverse temperatures up to four decimal places. Comparing the sum of squares with the benchmark sum of squares $S = 0.38142$ shows that all these sets are close to optimal in this example.

another version, the temperatures are set geometrically. The sampler is run for $N = 100\,000$ iterations (not including the burn-in of $10\,000$ iterations). The final long run shows that the geometric and the optimal temperature scheme give similar acceptance rates (0.09 and 0.08 respectively). Furthermore, both schemes yield a similar quality of mixing measured by the integrated autocorrelation times (see Table 7-5). As the mixing is similar, it is sufficient to display the histograms and traceplots of the sampled random variables for one of the schemes, here the optimal scheme (see Figures 7-16 to 7-18). Tempered transitions mixes well between labels. The histograms of the sampled component weights, means and variances resemble each other, respectively. It is also interesting to see the improvement from the $n = 16$ geometric temperatures scheme used in the previous robustness test of the interpolation curve to the $n = 60$ geometric temperatures scheme (see Tables 7-5 and 7-6). The gain in mixing is worth the fourfold computational cost: the estimated integrated autocorrelation time of the $n = 60$ temperatures scheme is on average 16 times better in the mixing between weights, 160 times better in the mixing between means and 27 times better in the mixing between variances than the $n = 16$ temperatures scheme. The acceptance rate (0.13) for $n = 16$ geometric temperatures is larger than the acceptance rate (0.08) for

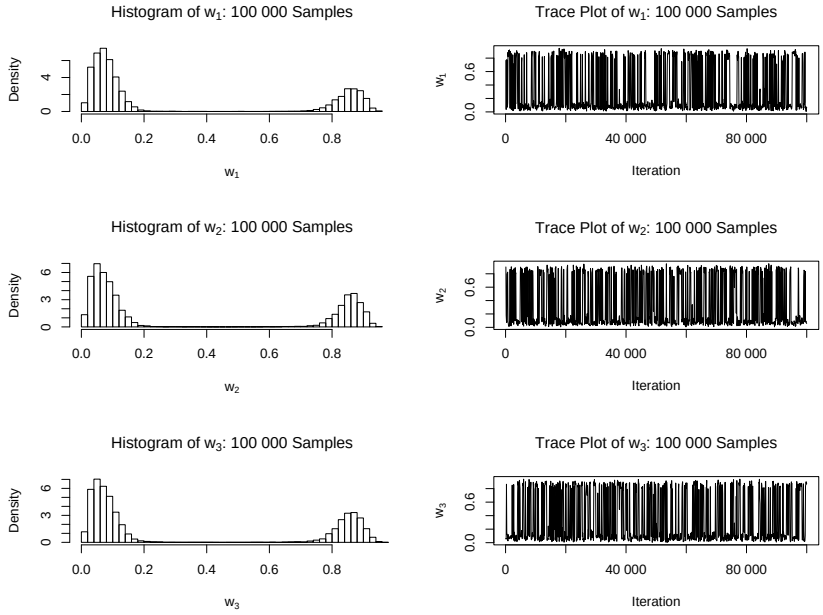

Figure 7-16: The histograms and traceplots of the posterior weights w_1, w_2, w_3 at the target temperature obtained by running tempered transitions based on 60 optimal temperatures between $\beta_{\min} = \frac{1}{8}$ and $\beta_0 = 1$ for 100 000 iterations. The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

$n = 60$. This is thus a good example to show that the acceptance rate is not the best measure for efficiency since the algorithm ($n = 60$) giving the smaller acceptance rate is, despite its higher cost, the more efficient one.

7.10 Summary

In this chapter, we have tested the tuning technique developed in Chapter 5 on a real application to show how we can proceed in practice. First, we had to decide on the way of tempering the distribution. We had to be careful not to design an improper algorithm. We have shown that, in Bayesian problems, tempering the likelihood while leaving the prior unchanged always defines proper distributions. The next step was to find an appropriate hottest distribution which should not be hotter than necessary as discussed in Chapter 5. Since the mixing at the hottest temperature is crucial for the mixing of tempered transitions, care was taken to construct an efficient standard MCMC sampler at that temperature. After that, we had to approximate the unknown curve $g(\beta)$ for the optimisation. Our strategy was to take the sample from the

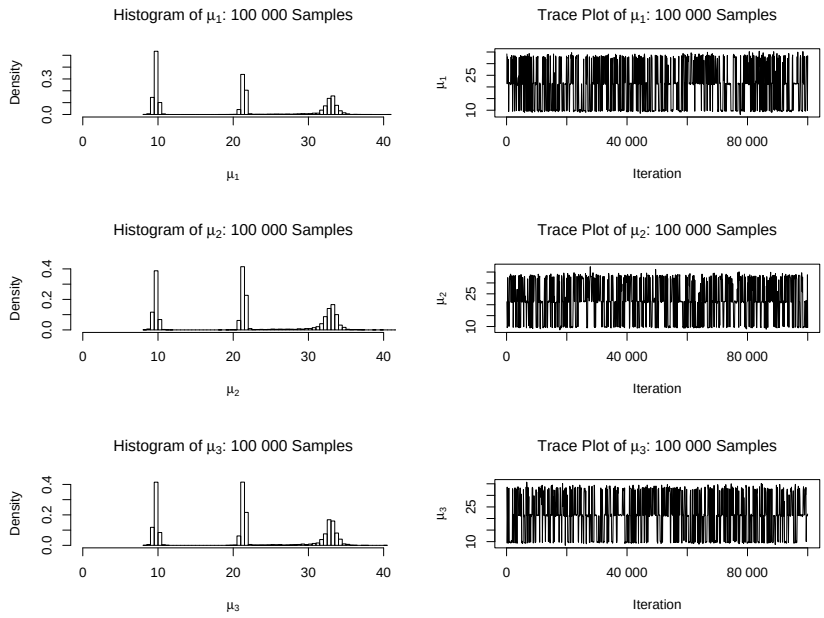

Figure 7-17: The histograms and traceplots of the posterior means μ_1, μ_2, μ_3 at the target temperature obtained by running tempered transitions based on 60 optimal temperatures between $\beta_{\min} = \frac{1}{8}$ and $\beta_0 = 1$ for 100 000 iterations. The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

hottest distribution which had to be generated anyway to check the mixing at that temperature and to use this sample to estimate some curve values and its derivatives at few β values between the hottest and the target temperature. Based on these anchor points, the curve could then be interpolated and the optimisation could be carried out as usual. We could dismiss worries about the accuracy of this procedure by showing that the minor variability in the curve approximation did not affect the optimisation results. In this example, the interpolated curve resembled closely the shape of the geometric model curve $g(\beta) = \frac{1}{2\beta}$ so that the optimal scheme and the geometric scheme lay close together. The final tempered transitions run confirmed that optimal and geometric scheme are of similar efficiency in this example. For the tempered transitions runs used in this chapter, we also had to specify the step sizes at each temperature. This was done by carrying out short preliminary runs with different candidate plans and choosing the one that was mixing best according to the graphical output (traceplots and histograms).

To close the fixed-dimensional work on tempered transitions, let us remark

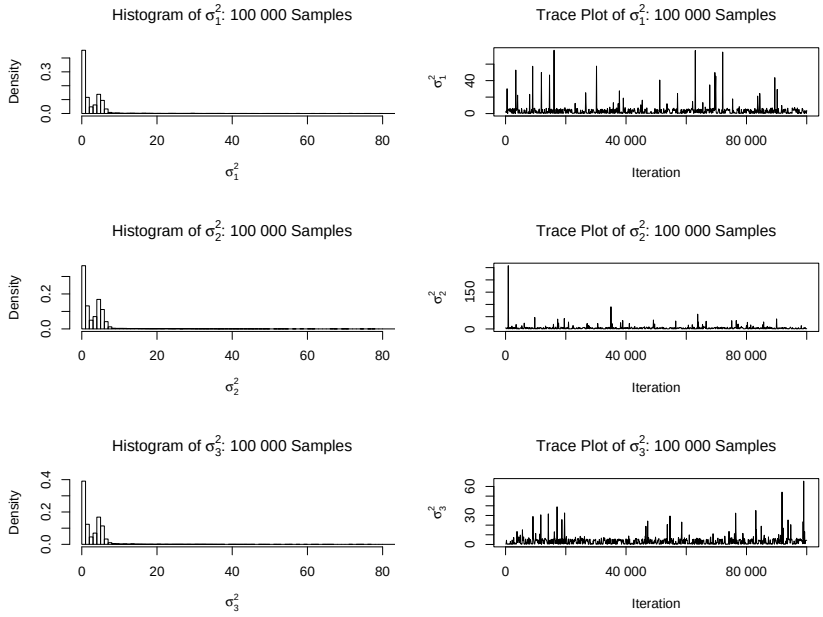

Figure 7-18: The histograms and traceplots of the posterior variances σ_1^2, σ_2^2, σ_3^2 at the target temperature obtained by running tempered transitions based on 60 optimal temperatures between $\beta_{\min} = \frac{1}{8}$ and $\beta_0 = 1$ for 100 000 iterations. The traceplots show every 100th sample. The symmetry between histograms indicates convergence in label switching.

that the tuning technique is more of a guidance than an obligation. Given that it is based on idealising assumptions that are not met in practice, we can probably decide on a sufficiently efficient temperature scheme by inspecting a plot of the estimated anchor points for the curve. Such a plot will already give us a rough idea of the curve's shape. If the shape resembles the model curve $g(\beta) = \frac{1}{2\beta}$, we can use the geometric scheme. If it is almost a straight line, a linear spacing should be suitable. And if it is clearly concave, then an anti-geometric scheme will be reasonable. On the other hand, once the code for the curve approximation and the optimisation is written, it can be used for any set of anchor points and for any approximated curve. In that case, the optimisation causes little extra effort so that we can only win by using it.

Having seen how powerful tempered transitions is in mode jumping in fixed dimension, we are interested in finding out whether it can be applied in variable dimension, and if so, whether it has a similar benefit. Let us start with reviewing the standard MCMC methods and the mode-jumping methods designed for variable dimension in the following chapter.

		60 Temperatures			
temperature scheme	k	$\hat{\tau}(w_k)$	$\hat{\tau}(\mu_k)$	$\hat{\tau}(\sigma_k^2)$	time in h
optimal	1	236.7	171.3	35.5	26
	2	160.5	142.9	29.8	
	3	186.5	164.7	31.4	
geometric	1	203.3	179.5	32.2	26
	2	215.9	170.6	28.4	
	3	229.4	159.3	33.3	

Table 7-5: The estimated integrated autocorrelation times of the component weights w_1, w_2, w_3, the component means μ_1, μ_2, μ_3 and the component variances σ_1^2, σ_2^2, σ_3^2 obtained by tempered transitions based on (a) 60 optimal temperatures and (b) 60 geometric temperatures and the computing time of tempered transitions in hours. The autocorrelation times are estimated with respect to the group mean, e.g. $\hat{\tau}(\mu_k)$ with respect to $\bar{\mu} = \frac{1}{K}\sum_{k=1}^{K}\bar{\mu}_k$ etc. The optimal and the geometric temperature scheme perform equally well.

		16 Temperatures			
temperature scheme	k	$\hat{\tau}(w_k)$	$\hat{\tau}(\mu_k)$	$\hat{\tau}(\sigma_k^2)$	time in h
geometric	1	3638.2	6146.1	1115.4	7
	2	3891.3	5685.4	1166.5	
	3	2704.8	6193.4	230.9	

Table 7-6: The estimated integrated autocorrelation times of the component weights w_1, w_2, w_3, the component means μ_1, μ_2, μ_3 and the component variances σ_1^2, σ_2^2, σ_3^2 obtained by tempered transitions based on 16 geometric temperatures and the computing time of tempered transitions in hours. The efficiency is in this example worse than the one of the run with 60 geometric temperatures (Table 7-5). The smaller computing time cannot make up for the higher autocorrelation times.

Chapter 8

Mode Jumping Methods in Variable Dimension: a Review

8.1 Introduction

In some statistical applications, the model is of variable dimension. For example, in the mixture model $\sum_{k=1}^{K} w_k \, p_k(y|\theta_k)$, the number K of components may be unknown. In this case, we often want to explore the uncertainty surrounding K and therefore model it as a random variable. The resulting model is of variable dimension because the number of component weights and parameters varies with the value K. The corresponding posterior distribution tends to be complex so that we often employ trans-dimensional MCMC methods to learn about it. In trans-dimensional MCMC, it is hard to jump between dimensions if the modes in the current dimension do not have "image" modes in the proposed dimension to which the sampler could be easily directed. In this case, the dimension-swapping proposals are likely to land in low-probability areas and are therefore often rejected.

As trans-dimensional MCMC often involves the transformation of measures, we will start with some measure theory (Section 8.2). After that, we will describe the most common trans-dimensional MCMC method "reversible jump MCMC" (RJMCMC), first in its general framework (Section 8.3) and then in its standard form (Section 8.4). We will also verify that standard RJMCMC satisfies detailed balance (Section 8.5), and introduce some informal notation (Section 8.6) which is often used in the literature. Finally, we will review alternatives and further developments of RJMCMC which are often designed to improve the mixing between modes of different dimension (Section 8.7).

8.2 Transforming measures

In trans-dimensional MCMC, a new proposal is often drawn from some distribution centred at the projection of the current state. Since the projection is part of the proposal mechanism, we have to account for it by an appropriate transformation of measures when specifying acceptance probabilities. We will therefore review some transformation theorems (see for example Section II.19 in Bauer 2001), which will be in particular helpful for verifying detailed balance in RJMCMC later.

To be able to transform measures, we need an $\mathcal{A}^*$-$\mathcal{A}'$-measurable mapping

$$T:\ (\Omega^*, \mathcal{A}^*) \to (\Omega', \mathcal{A}')$$

between the measure space $(\Omega^*, \mathcal{A}^*, \mu^*)$ and the measurable space $(\Omega', \mathcal{A}')$. The term $\mathcal{A}^*$-$\mathcal{A}'$-measurable means that

$$T^{-1}(A') \in \mathcal{A}^* \qquad \text{for every } A' \in \mathcal{A}'.$$

Such a transformation induces an image measure

$$\mu' := T(\mu^*)$$

defined by

$$\mu':\ A' \mapsto \mu^*\big(T^{-1}(A')\big),$$

which means that every set $A' \in \mathcal{A}'$ is assigned the μ^*-measure of its pre-image $T^{-1}(A')$. We may wish to integrate over an $\mathcal{A}'$-measurable numerical function g' on Ω'. We can do this if and only if $g' \circ T$ is μ^*-integrable, in which case

$$\int_{\Omega'} g' \,\mathrm{d}T(\mu^*) = \int_{\Omega^*} g' \circ T \,\mathrm{d}\mu^*.$$

This general transformation theorem for integrals also holds if we want to integrate g' over an $\mathcal{A}'$-measurable subset $G' \subset \Omega'$ since then also $(g'\,\mathbb{1}_{G'})$ is an $\mathcal{A}'$-measurable numerical function so that

$$\begin{aligned}
\int_{G'} g' \,\mathrm{d}T(\mu^*) &= \int_{\Omega'} (g'\,\mathbb{1}_{G'})\ \mathrm{d}T(\mu^*) \\
&= \int_{\Omega^*} (g'\,\mathbb{1}_{G'}) \circ T \,\mathrm{d}\mu^* \\
&= \int_{\Omega^*} (g' \circ T)\ \mathbb{1}_{T^{-1}(G')} \,\mathrm{d}\mu^* \\
&= \int_{T^{-1}(G')} g' \circ T \,\mathrm{d}\mu^*.
\end{aligned} \tag{8.1}$$

Note that we have to be careful when applying this result in reverse order. We can only express the integral $\int_{G^*} g' \circ T \, \mathrm{d}\mu^*$ over a set $G^* \in \mathcal{A}^*$ in terms of a μ'-integral if there exists a set $G' \in \mathcal{A}'$, of which $G^* = T^{-1}(G')$ is the pre-image, which is not always the case. For example, if $T(x) = x^2$ and $G^* = (5, 10)$, then $G^* = (5, 10)$ is mapped onto $G' = (25, 100)$, but G^* is not the pre-image of G', which is $T^{-1}(G') = (-5, -10) \cup (5, 10)$. Hence, $\int_{(5,10)} g' \circ T \, \mathrm{d}\mu^*$ cannot be expressed with respect to $\mathrm{d}T(\mu^*)$.

A special case of the transformation theorem is that for Lebesgue integrals. Suppose H^* and H' are open subsets in $\mathbb{R}^d$ and the transformation $\varphi : H^* \to H'$ is a diffeomorphism of H^* onto H'. [Recall that a diffeomorphism is a bijective continuously differentiable mapping, whose inversion is also continuously differentiable.] The theorem then says that a numerical function h' on H' is λ^d-integrable if and only if $h' \circ \varphi \; |\det \varphi'|$ is λ^d-integrable over H^*, in which case

$$\int_{H'} h' \, \mathrm{d}\lambda^d = \int_{H^*} (h' \circ \varphi) \; |\det \varphi'| \, \mathrm{d}\lambda^d \tag{8.2}$$

where φ' denotes the derivative of φ and $\det \varphi'$ its determinant. Note that the term $|\det \varphi'|$ appears because the transformed measure $\varphi^{-1}(\lambda^d)$ can be written in terms of the original measure λ^d by

$$\mathrm{d}\varphi^{-1}(\lambda^d) = |\det \varphi'| \, \mathrm{d}\lambda^d. \tag{8.3}$$

The equation (8.3) helps to show that the transformation theorem for Lebesgue integrals (8.2) is a special case of the previous result (8.1) applied to $T := \varphi^{-1}$, $\mu^* := \lambda^d$, $g' := (h' \circ \varphi)$, $G' = H^*$ and $T^{-1}(G') = H'$:

$$\begin{aligned} \int_{H^*} (h' \circ \varphi) \; |\det \varphi'| \, \mathrm{d}\lambda^d &= \int_{H^*} (h' \circ \varphi) \, \mathrm{d}\varphi^{-1}(\lambda^d) \\ &= \int_{H'} (h' \circ \varphi) \circ \varphi^{-1} \, \mathrm{d}\lambda^d \\ &= \int_{H'} h' \, \mathrm{d}\lambda^d. \end{aligned}$$

8.3 General RJMCMC

Green (1995, 2003) proposed the general reversible jump MCMC (RJMCMC) framework to allow sampling from some distribution π defined on a measure space $(\Omega, \mathcal{A}, \mu)$ where the sample space $\Omega = \bigcup_{K \in M} \Omega_K$ is determined by countably many subspaces $\{\Omega_K\}_{K \in M}$ of varying dimension. When discussing RJMCMC, it is helpful to have a particular application in mind.

We will explain RJMCMC in the context of variable-dimensional mixture modelling. Let us consider the mixture of univariate normal distributions $\sum_{k=1}^{K} w_k \, N(\mu_k, \sigma_k^2)$. For short, we will refer to the parameter vector of the K-component mixture model by

$$\theta_K := \left\{ w_k, \mu_k, \sigma_k^2 \right\}_{k \in \underline{K}} .$$

As the weights $\{w_k\}_{k \in \underline{K}}$ are constrained to sum to one, i.e. $\sum_{k=1}^{K} w_k = 1$, one of the weights is a dummy variable so that $\theta_K \in \mathbb{R}^{3K-1}$. For a given $K \in M$, let $p\left(\theta_K | K\right)$ denote the prior density (with respect to the Lebesgue measure on $\mathbb{R}^{3K-1}$), and let $p\left(y | K, \theta_K\right)$ denote the likelihood for the data $y := \{y_j\}_{j \in \underline{n}}$. Since we want to vary K, we also need a prior distribution $p\left(K\right)$ on the candidate models $K \in M$. To clarify which model we currently consider, we will include the model indicator when defining each subspace by $\Omega_K = \{K\} \times \Theta_K$ where $\Theta_K = \mathbb{R}^{3K-1}$. The corresponding submeasure for measurable $A = \{K\} \times A_K \subset \Omega_K$ is

$$\begin{aligned} \mu_K\left(A\right) &= \mu_K\left(\{K\} \times A_K\right) \\ &:= \lambda^{3K-1}\left(A_K\right) \end{aligned} \tag{8.4}$$

where λ^{3K-1} is the $(3K-1)$-dimensional Lebesgue measure. We can now define the measure μ on the combined space $\Omega = \bigcup_{K \in M} \Omega_K$ by

$$\mu\left(A\right) := \sum_{K \in M} \mu_K\left(A \cap \Omega_K\right) \qquad \text{for measurable } A \subset \Omega \, . \tag{8.5}$$

The joint posterior distribution with respect to μ is then given by the density

$$p\left(K, \theta_K | y\right) = \frac{p\left(K\right) p\left(\theta_K | K\right) p\left(y | K, \theta_K\right)}{\sum_{J \in M} \int p\left(J\right) p\left(\theta_J | J\right) p\left(y | J, \theta_J\right) \lambda^{3J-1}\left(\mathrm{d}\theta_J\right)} .$$

Using the full notation, we will express the joint posterior up to proportionality by

$$\begin{aligned} &p\left(K, \left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}} \middle| \left\{y_j\right\}_{j \in \underline{n}}\right) \\ &\qquad \propto p\left(K\right) \, p\left(\left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}} \middle| K\right) \, p\left(\left\{y_j\right\}_{j \in \underline{n}} \middle| K, \left\{w_k, \mu_k, \sigma_k^2\right\}_{k \in \underline{K}}\right) . \end{aligned}$$

Within each subspace Ω_K, we can sample by MCMC as usual. For moving between the subspaces, we need trans-dimensional MCMC. Green's idea is to introduce countably many move types m, which propose switching between subspaces via the corresponding proposal distribution q_m. In the mixture modelling example, we may move between two adjacent models by creating a new component (when moving from Ω_K to Ω_{K+1}) or by deleting one of the components (when moving from Ω_{K+1} to Ω_K in the reverse step); another way may be to split one component into two (when moving from

Ω_K to Ω_{K+1}) or by merging two components into one (when moving from Ω_{K+1} to Ω_K in the reverse step). RJMCMC considers each of the possible reversible moves as a single move type m. That means if K can take any value in $\{1, \ldots, K_{\max}\}$, then we have in total $2K_{\max}$ move types. We will label the birth-death move between the K-component and the $(K+1)$-component model as type $m = 2\,K - 1$, $K = 1, \ldots, K_{\max}$, and the split-combine move between the K-component and the $(K+1)$-component model as type $m = 2\,K$, $K = 1, \ldots, K_{\max}$. In RJMCMC, each move type m may be chosen with a probability depending on the current state $x \in \Omega$. [In the mixture modelling example, x stands for $x = (K, \theta), \theta_K \in \Theta_K$.] It is interesting that the probabilities of choosing each move type m do not have to sum to one so that $0 < \sum_m q_m(x, \Omega) \leq 1$; hence, with probability $(1 - \sum_m q_m(x, \Omega))$, no move is attempted at all. Moreover, as in the above example, not every move type m may be available for every current state $x \in \Omega$ so that $q_m(x, \Omega) = 0$ for some, perhaps many m. If $\Omega_K = \{K\} \times \mathbb{R}^{d_K}$, the RJMCMC transition kernel can be expressed by

$$P(x, B) = \sum_m \int_B q_m(x, \mathrm{d}x')\, \alpha_m(x, x') + s(x)\, \mathbb{1}_{\{x \in B\}}, \qquad \text{for all } B \in \mathcal{A},$$

where

$$s(x) := \sum_m \int_\Omega q_m(x, \mathrm{d}x')\ [1 - \alpha_m(x, x')] + \left(1 - \sum_m q_m(x, \Omega)\right)$$

defines the probability of remaining in x either because the proposed state is rejected or because no move type is chosen so that the detailed balance condition

$$\int_A \int_B \pi(\mathrm{d}x) P(x, \mathrm{d}x') = \int_B \int_A \pi(\mathrm{d}x') P(x', \mathrm{d}x)$$

takes the form

$$\begin{aligned} \sum_m \int_A \int_B \pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')\, \alpha_m(x, x') + \int_{A \cap B} \pi(\mathrm{d}x)\, s(x) \\ = \sum_m \int_B \int_A \pi(\mathrm{d}x')\, q_m(x, \mathrm{d}x)\, \alpha_m(x', x) + \int_{B \cap A} \pi(\mathrm{d}x')\, s(x'). \end{aligned}$$

This condition is satisfied if, for every move type m, the following detailed balance holds:

$$\int_A \int_B \pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')\, \alpha_m(x, x') = \int_B \int_A \pi(\mathrm{d}x')\, q_m(x', \mathrm{d}x)\, \alpha_m(x', x) \qquad \text{for all } A, B \in \mathcal{A}.$$

If $\pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')$ has a finite density

$$f_m(x, x') = \pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')$$

with respect to a symmetric measure ν_m on $\mathcal{A} \otimes \mathcal{A}$, the detailed balance condition for every move type can be achieved by the usual Metropolis-Hastings acceptance probability

$$\begin{aligned}\alpha_m(x, x') &= \min\left\{1, \frac{f_m(x', x)}{f_m(x, x')}\right\} \\ &= \min\left\{1, \frac{\pi(\mathrm{d}x')\, q_m(x', \mathrm{d}x)}{\pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')}\right\}\end{aligned}$$

giving

$$\begin{aligned}\int_A \int_B \pi(\mathrm{d}x)\, q_m(x, \mathrm{d}x')\, \alpha_m(x, x') &= \int_A \int_B \nu_m(\mathrm{d}x, \mathrm{d}x')\, f_m(x, x')\, \min\left\{1, \frac{f_m(x', x)}{f_m(x, x')}\right\} \\ &= \int_B \int_A \nu_m(\mathrm{d}x', \mathrm{d}x)\, f_m(x', x)\, \min\left\{1, \frac{f_m(x, x')}{f_m(x', x)}\right\} \\ &= \int_B \int_A \pi(\mathrm{d}x')\, q_m(x', \mathrm{d}x)\, \alpha_m(x', x)\end{aligned}$$

as required. Indeed, RJMCMC extends the Metropolis-Hastings framework to variable-dimensional sample spaces.

RJMCMC also introduces a new way of combining MCMC kernels: it chooses a move type randomly, but conditional on the current state. In the above description of RJMCMC, the proposal distribution $q_m(x, x')$ includes the probability of picking move type m when in x. For clarity, we will separate the probability $p(m|x)$ of choosing move type m when in x and the probability $q(x \to x'|m)$ of moving from x to x' under move type m so that

$$q_m(x, x') = p(m|x)\, q(x \to x'|m).$$

In the new notation, the move from x to x' under move type m is then accepted with probability

$$\alpha(x \to x'|m) = \min\left\{1, \frac{\pi(x')\, p(m|x')\, q(x' \to x|m)}{\pi(x)\, p(m|x)\, q(x \to x'|m)}\right\}. \tag{8.6}$$

This acceptance probability guarantees that the combination of RJMCMC kernels satisfies detailed balance, namely that

$$\begin{aligned}\pi(x) \sum_m p(m|x)\, q(x \to x'|m)\, \alpha(x \to x'|m) &= \sum_m \pi(x)\, p(m|x)\, q(x \to x'|m)\, \alpha(x \to x'|m) \\ &= \sum_m \pi(x')\, p(m|x')\, q(x' \to x|m)\, \alpha(x' \to x|m) \\ &= \pi(x') \sum_m p(m|x')\, q(x' \to x|m)\, \alpha(x' \to x|m).\end{aligned}$$

This way of combining MCMC kernels is new. So far, MCMC kernels have only been combined in an independent random order. Applying MCMC kernels in

independent random order can be seen as a special case of the new framework in which

$$p(m|x) = p(m).$$

Plugging $p(m|x) = p(m)$ into the acceptance probability (8.6) gives then the standard Metropolis-Hastings acceptance probability

$$\alpha(x \to x'|m) = \min\left\{1, \frac{\pi(x')\, q(x' \to x|m)}{\pi(x)\, q(x \to x'|m)}\right\}.$$

8.4 A common class of RJMCMC samplers

Green (1995) develops a class of standard RJMCMC algorithms which can be applied in the common case that each subspace $\Omega_K = \{K\} \times \Theta_K$ of the combined state space $\Omega = \bigcup_K \Omega_K$ is defined on the real parameter space $\Theta_K = \mathbb{R}^{d_K}$ of model-dependent dimension d_K. In particular, standard RJMCMC can be used when sampling from the Bayesian mixture model $\sum_{k=1}^{K} w_k\, N(\mu_k, \sigma_k^2)$ with an unknown number of components. In the previous section, we have seen that a move type m is chosen dependent on the current state. Standard RJMCMC checks in which subspace the current state $x \in \Omega$ lies. Suppose x comes from Ω_I, then the move type m which leads into the subspace Ω_J is chosen with probability $q_m(I, J)$. This notation accounts for the possibility that there are several move types m for jumping between the sub-spaces Ω_I and Ω_J. Recall that each move type m defines a reversible move between Ω_I and Ω_J. That means if we choose move type m when we are in Ω_J, this move will lead us to Ω_I so that we denote the probability of picking m when in Ω_J by $q_m(J, I)$. For ease of presentation, let us now assume that there is only one move type for jumping between Ω_I and Ω_J so that we will drop the index m when denoting move type specific probabilities.

Green suggests constructing the proposal from Ω_I to Ω_J with the help of auxiliary variables and transformation of variables as appropriate. For instance, when moving from the K-component mixture model to the $(K+1)$-component mixture model, we may propose adding a $(K+1)$th component to the current state by drawing a new weight w_{K+1}, a new mean μ_{K+1} and a new variance σ_{K+1}^2 independently from some proposal distribution (Richardson and Green 1997). Since the component weights are constrained to sum to one, we can adjust the old weights w_k, $k = 1, \ldots, K$, by multiplying them by $(1 - w_{K+1})$. Hence, the proposal consists of a random move (drawing

new variables) and a deterministic move (adjusting weights). Richardson and Green consider the extra variables as auxiliary variables and the deterministic step as a transformation of both the current (K-component) state and the auxiliary variables. Bearing this application in mind may help to understand the standard RJMCMC algorithm (Green 1995): after deciding to try a jump from the current space Ω_I to Ω_J with probability $q(I, J)$, we first draw an independent continuous auxiliary variable $u \in \Omega_U$ from some distribution $q_U(u)$ and then apply a diffeomorphism t that transforms (θ, u) into (θ', u'),

$$t: \ (\theta, u) \mapsto \Big(\theta'(\theta, u), u'(\theta, u)\Big).$$

This transformation yields the proposed model parameter $\theta' \in \Theta_J$ so that the proposed state is $x' = (J, \theta') \in \Omega_J$. The variable $u' \in \Omega_{U'}$ is an independent continuous auxiliary variable with distribution $q_{U'}(u')$. All the variables involved in the transformation must satisfy the dimension-matching condition

$$\dim(\theta) + \dim(u) = \dim(\theta') + \dim(u').$$

The standard RJMCMC proposal follows a proposal density $q_{IJ}(x, x')$ which, at the moment, can only be expressed in terms of (θ, u):

$$q_{IJ}\Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big) = \begin{cases} q(I, J)\, q_U(u) & \text{if } \Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big) \in \Omega_I \times \Omega_J, \\ 0 & \text{otherwise.} \end{cases} \tag{8.7}$$

The reverse move starts in $x' \in \Omega_J$. First we decide to try to jump from Ω_J to Ω_I with probability $q(J, I)$. Then we generate an independent auxiliary variable u' from some distribution $q_{U'}(u')$ and apply the inverse transformation

$$t^{-1}: \ (\theta', u') \mapsto \Big(\theta(\theta', u'), u(\theta', u')\Big)$$

to obtain (θ, u) from (θ', u'). Finally, we set the proposal state equal to $x = (I, \theta)$. The proposal density $q_{JI}(x', x)$ of the reverse move is thus given by

$$q_{JI}\Big(\Big(J, \theta'\Big), \Big(I, \theta(\theta', u')\Big)\Big) = \begin{cases} q(J, I)\, q_{U'}(u') & \text{if } \Big(\Big(J, \theta'\Big), \Big(I, \theta(\theta', u')\Big)\Big) \in \Omega_J \times \Omega_I, \\ 0 & \text{otherwise.} \end{cases} \tag{8.8}$$

Green defines the acceptance probability of the jump from $x = (I, \theta)$ to $x' = (J, \theta')$ by

$$\alpha_{IJ}\Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big) = \min\left\{1, \frac{\pi\Big(J, \theta'\,(\theta, u)\Big)\, q_{JI}\Big(\Big(J, \theta'(\theta, u)\Big), \Big(I, \theta\Big)\Big)\, d(\theta, u)}{\pi\Big(I, \theta\Big)\, q_{IJ}\Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big)}\right\} \tag{8.9}$$

where $d(\theta, u)$ denotes the absolute value of the determinant of the Jacobian matrix $\frac{\partial(\theta', u')}{\partial(\theta, u)}$ of the transformation t, which is a function of (θ, u):

$$d(\theta, u) := \left| \det \left(\frac{\partial(\theta', u')}{\partial(\theta, u)} \right) \right|.$$

Similarly, the reverse jump from $x' = (J, \theta')$ to $x = (I, \theta)$ is accepted with probability

$$\begin{aligned} &\alpha_{JI}\Big(\Big(I, \theta\Big), \Big(J, \theta(\theta', u')\Big)\Big) \\ &\quad = \min\left\{1, \frac{\pi\Big(I, \theta\left(\theta', u'\right)\Big)\, q_{IJ}\Big(\Big(I, \theta(\theta', u')\Big), \Big(J, \theta'\Big)\Big)}{\pi\Big(J, \theta'\Big)\, q_{JI}\Big(\Big(J, \theta'\Big), \Big(I, \theta(\theta', u')\Big)\Big)\, d\Big(\theta\left(\theta', u'\right), u\left(\theta', u'\right)\Big)}\right\} \end{aligned} \tag{8.10}$$

where $d\left(\theta(\theta', u'), u(\theta', u')\right) = \left| \det \left(\frac{\partial(\theta', u')}{\partial(\theta, u)} \right) \right|$ stands for the absolute value of the Jacobian determinant as above, but this time expressed in terms of (θ', u').

8.5 Detailed balance

It remains to verify that the Markov chain defined by the reversible jump between Ω_I and Ω_J described in Section 8.4 satisfies detailed balance for all $A \times B \in \mathcal{A} \otimes \mathcal{A}$ where $\mathcal{A}$ is the σ-algebra of the combined state space $\Omega = \bigcup_K \Omega_K$. Actually, the jump is defined on $\Theta_I \times \Omega_U$ and its reverse on $\Theta_J \times \Omega_{U'}$, so that a first step towards the ultimate goal is to check that for

$$A_I \times U := \Big\{(\theta, u) \in \Theta_I \times \Omega_U : \ (I, \theta) \in A \text{ and } \Big(J, \theta'(\theta, u)\Big) \in B\Big\}$$

and

$$B_J \times U' := \Big\{(\theta', u') \in \Theta_J \times \Omega_{U'} : \ (J, \theta) \in B \text{ and } \Big(I, \theta(\theta', u')\Big) \in A\Big\},$$

the following equality holds:

$$\begin{aligned} &\int_{A_I \times U} \pi\Big(I, \theta\Big)\, q_{IJ}\Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big)\, \alpha_{IJ}\Big(\Big(I, \theta\Big), \Big(J, \theta'(\theta, u)\Big)\Big) \mathrm{d}\lambda \\ &\qquad = \int_{B_J \times U'} \pi\Big(J, \theta'\Big)\, q_{JI}\Big(\Big(J, \theta'\Big), \Big(I, \theta(\theta', u')\Big)\Big)\, \alpha_{JI}\Big(\Big(J, \theta'\Big), \Big(I, \theta(\theta', u')\Big)\Big) \mathrm{d}\lambda \end{aligned}$$

where λ denotes the Lebesgue measure. As the transformation $t(\theta, u)$ is a diffeomorphism, we can apply the transformation theorem for Lebesgue

integrals (8.2), which yields

$$
\begin{aligned}
&\int_{B_J \times U'} \pi\Big(J,\theta'\Big)\, q_{JI}\bigg(\Big(J,\theta'\Big),\Big(I,\theta(\theta',u')\Big)\bigg)\, \alpha_{JI}\bigg(\Big(J,\theta'\Big),\Big(I,\theta(\theta',u')\Big)\bigg)\mathrm{d}\lambda \\
&\overset{(8.2)}{=} \int_{A_I \times U} \pi\Big(J,\theta'(\theta,u)\Big)\, q_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)\, \alpha_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)\, \left|\det\left(\frac{\partial(\theta',u')}{\partial(\theta,u)}\right)\right| \mathrm{d}\lambda \\
&= \int_{A_I \times U} \pi\Big(J,\theta'(\theta,u)\Big)\, q_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)\, \alpha_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)\, d(\theta,u)\,\mathrm{d}\lambda \\
&= \int_{A_I \times U} \pi\Big(I,\theta\Big)\, q_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg)\, \alpha_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg)\mathrm{d}\lambda,
\end{aligned}
\tag{8.11}
$$

as required. Note that the last step follows from the definition of α_{IJ} (8.9) and α_{JI} (8.10). We have thus shown that detailed balance holds when x and x' are expressed in terms of (θ, u), e.g. on the space $\Theta_I \times \Omega_U$. Intuitively, detailed balance should also hold if x and x' are expressed by themselves, e.g. on the space $\Omega \times \Omega$. This shall be verified in the following. First let us shorten the notation by setting

$$
f_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg) := \pi\Big(I,\theta\Big)\, q_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg),
$$

which is equal to zero if $\Big(I,\theta\Big) \neq \Omega_I$ by definition of q_{IJ} (8.7), and similarly, by setting

$$
f_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg) := \pi\Big(J,\theta'(\theta,u)\Big)\, q_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)\, d(\theta,u),
$$

which is equal to zero if $\Big(J,\theta'(\theta,u)\Big) \neq \Omega_J$ by definition of q_{JI} (8.8). The acceptance probabilities can then be written by

$$
\alpha_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg) = \min\left\{1, \frac{f_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)}{f_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg)}\right\}
$$

and

$$
\alpha_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg) = \min\left\{1, \frac{f_{IJ}\bigg(\Big(I,\theta\Big),\Big(J,\theta'(\theta,u)\Big)\bigg)}{f_{JI}\bigg(\Big(J,\theta'(\theta,u)\Big),\Big(I,\theta\Big)\bigg)}\right\}.
$$

For the proof, we need the transformation

$$
\tilde{T}:\ \Theta_I \times \Omega_U \to \Omega \times \Omega
$$

defined by

$$
\tilde{T}:\ \{(\theta,u)\} \mapsto \Big\{\underbrace{\Big(I,\theta\Big)}_{=x}, \underbrace{\Big(J,\theta'(\theta,u)\Big)}_{=x'}\Big\} \cup \Big\{\underbrace{\Big(J,\theta'(\theta,u)\Big)}_{=x'}, \underbrace{\Big(I,\theta\Big)}_{=x}\Big\}
$$

to induce a symmetric measure $\nu = \tilde{T}(\lambda)$ on $\Omega \times \Omega$ as required in the general RJMCMC framework. Note that the transformation $\tilde{T}$ is not bijective so that we need the general transformation theorem when transforming integrals later. Further note that the transformed measure $\tilde{T}(\lambda)$ is equal to zero if $A \times B \not\subset (\Omega_I \times \Omega_J) \cup (\Omega_J \times \Omega_I)$ so that detailed balance holds on $\Omega \times \Omega$ if it holds on $(\Omega_I \times \Omega_J) \cup (\Omega_J \times \Omega_I)$. To verify the later, we have to check that

$$\int_{A\times B} f_{IJ}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) = \int_{B\times A} f_{JI}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda)$$

in the cases $A \times B \subset (\Omega_I \times \Omega_J)$ and $A \times B \subset (\Omega_J \times \Omega_I)$. The second case is easy because f_{IJ} is zero on $A \times B$ and f_{JI} is zero on $B \times A$ so that

$$\int_{A\times B} \underbrace{f_{IJ}}_{=0 \text{ by def.}}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) = \int_{B\times A} \underbrace{f_{JI}}_{=0 \text{ by def.}}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda)$$

as required. Detailed balance also holds in the first case where $A \subset \Omega_I$ and $B \subset \Omega_J$ because

$$\begin{aligned}
\int_{A\times B} f_{IJ}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) &= \int_{A\times B} f_{IJ}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) + \int_{B\times A} \underbrace{f_{IJ}}_{=0 \text{ by def.}}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) \\
&= \int_{(A\times B)\cup(B\times A)} f_{IJ}\, \alpha_{IJ}\, \mathrm{d}\tilde{T}(\lambda) \\
&\overset{(8.1)}{=} \int_{\tilde{T}^{-1}((A\times B)\cup(B\times A))} (f_{IJ}\, \alpha_{IJ}) \circ \tilde{T}\, \mathrm{d}\lambda \\
&= \int_{A_I\times U} (f_{IJ}\, \alpha_{IJ}) \circ \tilde{T}\, \mathrm{d}\lambda \\
&\overset{(8.11)}{=} \int_{A_I\times U} (f_{JI}\, \alpha_{JI}) \circ \tilde{T}\, \mathrm{d}\lambda \\
&= \int_{\tilde{T}^{-1}((A\times B)\cup(B\times A))} (f_{JI}\, \alpha_{JI}) \circ \tilde{T}\, \mathrm{d}\lambda \\
&\overset{(8.1)}{=} \int_{(A\times B)\cup(B\times A)} f_{JI}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda) \\
&= \int_{A\times B} \underbrace{f_{JI}}_{=0 \text{ by def.}}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda) + \int_{B\times A} f_{JI}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda) \\
&= \int_{B\times A} f_{JI}\, \alpha_{JI}\, \mathrm{d}\tilde{T}(\lambda).
\end{aligned}$$

Thus, we have seen that standard RJMCMC satisfies detailed balance on $\Omega\times\Omega$.

8.6 Informal notation

Having understood the standard RJMCMC algorithm, we can now follow the informal notation often used in the literature. In this notation, the dependencies between variables are dropped so that, for instance, the acceptance

probabilities are denoted by

$$\alpha_{IJ}(x, x') = \min\left\{1, \frac{\pi(x')\, q(J, I)\, q_{U'}(u')}{\pi(x)\, q(I, J)\, q_U(u)} \left|\det\left(\frac{\partial(\theta', u')}{\partial(\theta, u)}\right)\right|\right\}$$

and

$$\alpha_{JI}(x', x) = \min\left\{1, \frac{\pi(x)\, q(I, J)\, q_U(u)}{\pi(x')\, q(J, I)\, q_{U'}(u')} \left|\det\left(\frac{\partial(\theta', u')}{\partial(\theta, u)}\right)\right|^{-1}\right\}.$$

This notation also eases the implementation of RJMCMC: we can compute each term according to its own parametrisation, e.g. $\pi(x)$ in dependence of x and $\left|\det\left(\frac{\partial(\theta', u')}{\partial(\theta, u)}\right)\right|$ in dependence of (θ, u) because the value of a particular function does not change with the transformation.

8.7 Further developments

In a follow-up paper to Green (1995), Richardson and Green (1997) explored RJMCMC move types for varying the number of components in mixture modelling for univariate data. They suggested the pairs of inverse moves: adding versus deleting a component (birth-death move) and creating two components out of one versus merging two components into one (split-combine move). These move types can also be generalised for mixtures modelling multivariate data (Dellaportas and Papageorgiou 2006). Another approach to mixture modelling in variable dimension is to simulate the births and deaths of components as a continuous time Markov birth-death process that converges to the target distribution in equilibrium (Stephens 2000): the model parameters are seen as a marked point process where each point represents one of the components; births and deaths of components occur at an exponential rate. While the birth rate is constant, the death rate varies from component to component as it depends on the current parameter value of the component. The death rate of a particular component will be higher, the less the component contributes to the explanation of the data. Cappé, Robert and Rydén (2003) view this birth-death set-up as a special case of a broader framework based on Markov jump processes. In a Markov jump process, jumps from one state to another occur at random times. The time between jumps follows an exponential distribution whose rate depends on the current state. At the jump times the process moves from the current state to a new state through a Markov transition kernel satisfying detailed balance with respect to the target distribution. The advantage of this generalisation is that additional moves such as split-combine moves can be incorporated into the continuous-time MCMC

sampler. Cappé et al. show that there is a strong link between continuous-time MCMC and (discrete-time) RJMCMC: it is possible to construct a (discrete-time) RJMCMC sampler which simulates the target process at the equally spaced times $\left\{\frac{i}{n}\right\}_{i\in\mathbb{N}_0}$ such that the RJMCMC sampler converges to the continuous-time sampler as the lag between updates tends to zero, i.e. as $n \to \infty$. Furthermore, Cappé et al. test both methods on modelling mixtures of variable dimension and find that RJMCMC is three times as fast as the corresponding continuous-time sampler, and is thus the superior method.

The main drawback of RJMCMC is that proposal mechanisms may be inefficient because too little is known about how models of different dimension engage with each other. A simple proposal mechanism for jumps into a higher-dimensional space may augment the current state by a random draw for the missing parameter values and, if necessary, adjust this proposal to satisfy any constraints. However, although models may be nested, the corresponding modes may not be so that such a simple proposal mechanism is prone to direct moves into a low-probability area of the alternative model. Brooks, Giudici and Roberts (2003) face this problem when modelling autoregressive processes of unknown order, i.e. on modelling $X_t = \sum_{j=1}^{m} \psi_j X_{t-j} + \varepsilon_t$, where $\{\psi_j\}$ are the model parameters, m is the unknown order of the process, and ε_t is white noise, e.g. $\varepsilon_t \sim N(0, \sigma_\varepsilon^2)$. They propose to move from the current m-dimensional state $\{\psi_j\}_{j\in\underline{m}}$ to $\{\psi'_j\}_{j\in\underline{(m+1)}}$ by keeping the current values $\psi'_j = \psi_j$, $j \in \underline{m}$, and drawing the additional parameter from a normal distribution $\psi'_{m+1} \sim N(0, \sigma_\psi^2)$. The acceptance rate for this move type depends heavily on the variance of the normal proposal distribution. There is the danger that the variance is either too small or too large so that care needs to be taken when tuning the variance. As tuning will be difficult and time consuming in complex problems, Brooks et al. investigate how to construct more efficient move types. The basic idea is to centre the proposal for the higher-dimensional jump at the point in the higher-dimensional space defining a model that is identical to the lower-dimensional model. They call this point non-identifiability centre. In the autoregressive example, if the current state is $(\psi_1, \ldots, \psi_m)$, then the non-identifiability centre will be at $(\psi_1, \ldots, \psi_m, 0)$ with the non-identifiability arising from the identity

$$\sum_{j=1}^{m} \psi_j X_{t-j} + 0\, X_{t-(m+1)} + \varepsilon_t \equiv \sum_{j=1}^{m} \psi_j X_{t-j} + \varepsilon_t\,.$$

In other words, we cannot tell the new model $\sum_{j=1}^{m} \psi_j X_{t-j} + 0\, X_{t-(m+1)} + \varepsilon_t$ from the old model $\sum_{j=1}^{m} \psi_j X_{t-j} + \varepsilon_t$ because the new model can be written

as the old model by $\sum_{j=1}^{m} \psi_j X_{t-j} + 0\, X_{t-(m+1)} + \varepsilon_t = \sum_{j=1}^{m} \psi_j X_{t-j} + \varepsilon_t$. Actually, the problematic proposal mechanism, which retains the current parameters and adds a new parameter by $\psi'_{m+1} \sim N(0, \sigma_\psi^2)$, is already centred at the non-identifiability point. Therefore, Brooks et al. suggest further refinements of the proposal mechanism. One of these refined methods picks proposals from the close neighbourhood of the current state. For this, the step size of the proposal distribution is chosen dependent on the current state. It has to satisfy the following two conditions: first, the acceptance probability for a move from the current state $(\psi_1, \ldots, \psi_m)$ to its projected image $(\psi_1, \ldots, \psi_m, 0)$ is equal to one; second, the derivative of the logarithm of this acceptance probability is equal to zero. Brooks et al. report that this state-dependent proposal mechanism doubles the acceptance rate of the previous state-independent proposal mechanism with tuned proposal variance. To ensure that the RJMCMC algorithm based on the refined proposal mechanism mixes well, additional fixed-dimensional MCMC move types, which mix well within each model, need to be incorporated into the algorithm.

RJMCMC may also have poor acceptance rates if there are zero-probability-constraints in the sample space. In an object recognition example, Al-Awadhi, Hurn and Jennison (2004b) modelled the location, size and orientation of an unknown number of plant cells under the constraint that cells do not overlap. This already causes problems when sampling from the fixed-dimensional distribution. For instance, rotating a single cell may cause its overlapping with a neighbouring cell and thus the rejection of the move so that the sampler may be trapped in a certain cell configuration. Similarly, trans-dimensional moves will not be accepted if the proposed new cell overlaps with one of the existing cells. Hence, split-combine moves, that either split a cell into two or merges two neighbouring cells into one, may be better suited to move between dimensions. Also excursions over an unconstrained sample space as introduced by Hurn et al. (1999) improved the mixing. In a follow-up paper, Al-Awadhi, Hurn and Jennison (2004a) tried excursions over a tempered distribution. This solution gave the most satisfying results. As this solution involves ideas from tempering, we shall briefly present it here. To find a sensible proposal state x' in a higher-dimensional space than the current state x, first an auxiliary state z, which lies in the higher-dimensional space, is drawn from an RJMCMC proposal distribution, but no acceptance/rejection decision is made. This auxiliary state serves merely as a starting point for a sequence of k standard MCMC steps

(including acceptance/rejection), which satisfy detailed balance with respect to the tempered distribution. The final state x' of these k steps hopefully lies in the modal region of the target distribution, and a move from the current state x to the final state x' is considered with a probability that also depends on the intermediate state z. Suppose the current state x is of dimension K and the auxiliary state z and the proposal state x' are of dimension K' which is greater than K. Let $q(x, z)$ denote the trans-dimensional proposal distribution, and P denote the MCMC kernel for each of the k intermediate steps leading from z to x'. As the kernel P satisfies detailed balance with respect to the tempered conditional distribution $p_\beta(\cdot|K')$, detailed balance holds also for the kth iterate P^k, i.e.

$$p_\beta(z|K')\, P^k(z, x') = p_\beta(x'|K')\, P^k(x', z),$$

so that

$$\frac{P^k(x', z)}{P^k(z, x')} = \frac{p_\beta(z|K')}{p_\beta(x'|K')}.$$

The Metropolis-Hastings acceptance probability for the trans-dimensional move from x to x' via z simplifies then to

$$\begin{aligned}\alpha\,(x, z, x') &= \min\left\{1, \frac{p(x'|K')\, P^k(x', z)\, q(z, x)}{p(x|K)\, q(x, z)\, P^k(z, x')}\right\} \\ &= \min\left\{1, \frac{p(x'|K')\, p_\beta(z|K')\, q(z, x)}{p(x|K)\, q(x, z)\, p_\beta(x'|K')}\right\}\end{aligned}$$

so that detailed balance of the move from $x \in A$ to $x' \in C$ via $z \in B$ is satisfied:

$$\begin{aligned}&\int_A \mathrm{d}x \int_B \mathrm{d}z \int_C \mathrm{d}x'\, p_\beta(x|K)\, q(x, z)\, P^k(z, x')\, \alpha\,(x, z, x') \\ &\qquad = \int_C \mathrm{d}x' \int_B \mathrm{d}z \int_A \mathrm{d}x\, p_\beta(x'|K')\, P^k(x', z)\, q(z, x)\, \alpha\,(x', z, x)\,.\end{aligned}$$

Note that the reverse move from x' to x first moves from x' to z via the k MCMC steps before the trans-dimensional jump from dimension K' to K is executed. Al-Awadhi et al. deliberately choose not to incorporate another k MCMC steps at the lower dimension because, in their application, the reverse move already lands in modal area of the lower-dimensional model, so that another k mode finding steps are not required. However, if another application needed such moves, the algorithm could be adapted accordingly.

In a different object recognition problem of locating an unknown number of neural sources from electromagnetic measurements of the brain activity (Bertrand, Ohmi, Suzuki and Kado 2001), mixing between modes is achieved

by applying parallel tempering (Geyer 1991), originally called Metropolis-coupled MCMC, first in a fixed-dimensional problem, then in a variable-dimensional problem. In the fixed-dimensional toy problem, parallel tempering mixes far better than standard MCMC; it reduces the mean error of estimation significantly. For sampling from the variable-dimensional toy problem, parallel tempering is combined with RJMCMC. That means that $(n+1)$ RJMCMC chains are run in parallel, each satisfying detailed balance to a different distribution, which is commonly a tempered version of the target distribution; after each RJMCMC chain has been updated, the method randomly swaps states between adjacent chains. For simplification, let us assume again that each chain is a tempered version of the target distribution of the form

$$p_{\beta_i}(x) \propto \pi(x) \exp\left[-\beta_i\, h(x)\right], \qquad i = 0, 1, \ldots, n,$$

with $\beta_{\min} = \beta_n < \cdots < \beta_0 = 1$. All the tempered distributions are defined on the same combined space $\Omega = \bigcup_K \Omega_K$. Also swapping states of adjacent chains takes place on that combined space so that the state swap can be carried out as usual. With probability

$$\alpha = \min\left\{1, \frac{p_{\beta_i}(x_j) p_{\beta_j}(x_i)}{p_{\beta_i}(x_i) p_{\beta_j}(x_j)}\right\}$$

the current state x_i of the p_{β_i}-invariant chain becomes the new state of the p_{β_j}-invariant chain, while the current state x_j of the p_{β_j}-invariant chain becomes the new state of the p_{β_i}-invariant chain. Unfortunately, Bertrand et al. do not compare the parallel tempering RJMCMC algorithm with standard RJMCMC.

Jasra, Stephens and Holmes (2007b) also apply parallel tempering in variable dimension. To improve the mixing of each chain, they also incorporate move types borrowed from population-based MCMC so that chains can learn from each other without necessarily swapping states. They show in an example that parallel tempering mixes better than standard RJMCMC.

In conclusion, there is a great demand to improve jumping between modes of different dimension. As tempering ideas seem to perform well in RJMCMC, we will explore the benefits of applying tempered transitions based on RJMCMC steps when sampling from variable-dimensional models.

Chapter 9

Tempered Transitions in Variable Dimension

9.1 Introduction

In this chapter, we will explore the use of tempered transitions for sampling from a variable-dimensional distribution. First we will check the validity of the tempered transitions RJMCMC algorithm (Section 9.2). Then we will apply the method in variable-dimensional mixture modelling. For this, we will introduce a variable-component mixture model for the galaxy data (Section 9.3) and a birth-and-death move type for sampling from it (Section 9.4). After that, we will tune the tempered transitions RJMCMC algorithm and compare its performance to that of standard RJMCMC (Section 9.5).

9.2 Validity of tempered transitions RJMCMC

Tempered transitions RJMCMC samples from a variable-dimensional distribution p by carrying out the tempered transitions steps as before (see Section 4.2.2), but now with respect to variable-dimensional auxiliary distributions p_{β_i}, $i = 0, 1, \ldots, n$, and corresponding RJMCMC kernels T_{β_i}, $i = 0, 1, \ldots, n$. Verifying the validity of this method is straightforward. First, we have to check on a case-by-case basis that the auxiliary distributions p_{β_i}, $i = 0, 1, \ldots, n$, defined on the general state space $\Omega = \bigcup_K \Omega_K$ are proper. Second, the RJMCMC transition kernels T_{β_i}, $i = 0, 1, \ldots, n$, should satisfy detailed balance

$$p_{\beta_i}(x)\, T_{\beta_i}(x, x') = p_{\beta_i}(x')\, T_{\beta_i}(x', x) \qquad \forall\, x, x' \in \Omega$$

with respect to the corresponding auxiliary distribution p_{β_i}. This does not cause any problems if the kernels are standard RJMCMC kernels for which detailed balance always holds (see Section 8.5). If detailed balance is satisfied, the reversibility of the tempered transitions kernel follows automatically (see Section 4.2.3). Finally, we need to verify that the algorithm constructs an irreducible and aperiodic Markov chain, which can only be done on a case-by-case basis.

9.3 Variable-dimensional mixture model for the galaxy data

We will extend the fixed-component model for the galaxy data given in Section 7.3.3 to a variable-component model in which the number of components may vary between $K_{\min} = 1$ and $K_{\max} = 10$, which seems a reasonable choice (compare Figure 7-2). The new galaxy model is thus

$$\begin{aligned}
y_j \Big| K, \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}} &\sim \textstyle\sum_{k=1}^{K} w_k \, N(\mu_k, \sigma_k^2), && \forall\, j \in \underline{n}, \\
\{w_k\}_{k \in \underline{K}} \Big| K &\sim \mathit{Dirichlet}(1, \ldots, 1), \\
\mu_k \Big| K &\sim N(0, 1000), && \forall\, k \in \underline{K}, \\
\sigma_k^2 \Big| K &\sim \mathit{Inverse\ Gamma}(1, 1), && \forall\, k \in \underline{K}, \\
K &\sim U\{1, 2, \ldots, 10\}.
\end{aligned}$$

The corresponding distributions are given by

$$p\left(y_j \Big| K, \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right) \propto \sum_{k=1}^{K} w_k \, \left(\sigma_k^2\right)^{-\frac{1}{2}} \exp\left[-\frac{1}{2\,\sigma_k^2}\,(y_j - \mu_k)^2\right], \qquad \forall\, j \in \underline{n},$$

and

$$\begin{aligned}
p\left(\{w_k\}_{k \in \underline{K}} \Big| K\right) &= (K-1)!\, \mathbb{1}_{\{\sum_{k=1}^{K} w_k = 1\}}, && w_k \in [0,1], && \forall\, k \in \underline{K}, \\
p\left(\mu_k \Big| K\right) &= (2 \cdot 1000\,\pi)^{-\frac{1}{2}} \exp\left(-\tfrac{\mu_k^2}{2 \cdot 1000}\right), && \mu_k \in (-\infty, \infty), && \forall\, k \in \underline{K}, \\
p\left(\sigma_k^2 \Big| K\right) &= \left(\sigma_k^2\right)^{-1} \exp\left(-\tfrac{1}{\sigma_k^2}\right), && \sigma_k^2 \in (0, \infty), && \forall\, k \in \underline{K}, \\
p(K) &= \tfrac{1}{10}, && K \in \{1, 2, \ldots, 10\}.
\end{aligned}$$

The marginal distribution of a particular weight w_k given K is $\mathit{Beta}(1, K-1)$, i.e. $p(w_k|K) = (K-1)\,(1-w_k)^{K-2}$ for $0 \le w_k \le 1$. It is important to note that one of the weights, e.g. w_K, is a dummy variable as it is a function of the other weights, e.g.

$$w_K = 1 - \sum_{k=1}^{K-1} w_k.$$

In consequence, the dimension of the K-component model is $(3K-1)$ so that we, for example, do not integrate over the dummy variable when integrating over the prior or posterior distribution. This also means that we have to ignore the dummy variable when determining the Jacobian of a transformation used in the RJMCMC moves. To write down the posterior distribution, we need the joint prior distributions and the joint likelihood function. The variable-component model assumes that, a-priori, the component means and variances are independent of each other and of the weights, while the weights depend on each other so that the joint prior distributions are given by

$$\begin{aligned}
p\left(\{\mu_k\}_{k\in\underline{K}}\middle|K\right) &= \prod_{k=1}^{K} p\left(\mu_k\middle|K\right), \\
p\left(\{\sigma_k^2\}_{k\in\underline{K}}\middle|K\right) &= \prod_{k=1}^{K} p\left(\sigma_k^2\middle|K\right), \\
p\left(\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\middle|K\right) &= p\left(\{w_k\}_{k\in\underline{K}}\middle|K\right)\, p\left(\{\mu_k\}_{k\in\underline{K}}\middle|K\right)\, p\left(\{\sigma_k^2\}_{k\in\underline{K}}\middle|K\right), \\
p\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right) &= p\left(K\right)\, p\left(\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\middle|K\right).
\end{aligned}$$

While the prior $p\left(\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\middle|K\right)$ on the subspace Ω_K is defined with respect to the submeasure μ_K given in (8.4), the prior $p\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right)$ on the combined space Ω is defined with respect to the measure μ given in (8.5). The model assumes that the observations are conditionally independent so that the joint likelihood function is expressed by

$$p\left(\{y_j\}_{j\in\underline{n}}\middle|K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right) = \prod_{j=1}^{n} p\left(y_j\middle|K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right).$$

The resulting posterior (with respect to the measure μ) is known only up to proportionality:

$$p\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\middle|\,\{y_j\}_{j\in\underline{n}}\right) \propto p\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right)\, p\left(\{y_j\}_{j\in\underline{n}}\middle|K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right).$$

We also have to decide on the way of tempering the posterior distribution so that we can apply tempered transitions later. As before, we will only temper the likelihood part so that the auxiliary distributions are proportional to

$$\begin{aligned}
&p_\beta\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\middle|\,\{y_j\}_{j\in\underline{n}}\right) \\
&\quad\propto p\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right)\left[p\left(\{y_j\}_{j\in\underline{n}}\middle|K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right)\right]^\beta.
\end{aligned}$$

As shown in Section 7.5, this definition gives proper tempered distributions when $\beta\in(0,1)$. To make clear that the energy function h is well defined on Ω, we will write

$$h\left(K,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{K}}\right) := -\sum_{J\in M} \mathbb{1}_{\{K=J\}}\, \log\left[p\left(\{y_j\}_{j\in\underline{n}}\middle|J,\{w_k,\mu_k,\sigma_k^2\}_{k\in\underline{J}}\right)\right].$$

The tempered posterior distributions are then given by

$$p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right) \propto p\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}}\right) \exp\left[-\beta\, h\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}}\right)\right].$$

Having defined the posterior and the auxiliary distributions, we can now construct RJMCMC transition kernels with respect to these distributions. For simplicity, we will only use one trans-dimensional move type, namely the birth-and-death move. We will not implement any fixed-dimensional moves to be sure that any improvement in the mixing is due to the trans-dimensional move types in tempered transitions and not the fixed-dimensional ones because we know already that tempered transitions can lead to a vast improvement in fixed dimension.

9.4 Birth-and-death move

9.4.1 Positioning components

We will describe the birth-and-death move for the galaxy example from the point of view of how to implement it. For the implementation, we need a way of storing the components and rules about how this way changes with the birth or death of a component. In the galaxy example, the number K of components varies between $K_{\min} = 1$ and $K_{\max} = 10$ so that at most $K_{\max}$ positions are required to store one state of the RJMCMC chain. A pragmatic way of handling this variation is to introduce $K_{\max}$ constant positions $1, \ldots, K_{\max}$, each of which can either be free or occupied by one of the components. If position I is taken, the parameters of the component occupying I will be indexed by I so that the component parameters are called w_I, μ_I and σ_I^2. The important issue is how this allocation is affected by the birth-and-death move. A bad way of allocating variables would be to place the components of the K-component model always in the positions $1, \ldots, K$. In this case, a new component would be created at position $(K + 1)$, and, if the Ith component dies, then the components occupying the positions $(I + 1), \ldots, K$ would move down by one position so that all the positions $1, \ldots, (K - 1)$ would be filled again. This is a bad idea because it causes artificial label switching; we would lose track of the origin of parameter values very quickly so that we cannot tell whether the value at position I was created recently or whether it had simply been moved down from other positions by the artificial label switching process.

For $K_{\max} = 5$, this is illustrated in the following example:

$\underline{\clubsuit}\ \underline{\spadesuit}\ \underline{\heartsuit}\ \_\ \_$	
$\underline{\clubsuit}\ \underline{\heartsuit}\ \_\ \_\ \_$	death of $\spadesuit$
$\underline{\clubsuit}\ \underline{\heartsuit}\ \underline{\diamondsuit}\ \_\ \_$	birth of $\diamondsuit$

Due to the artificial repositioning, we may get the wrong impression that the RJMCMC algorithm proposed to delete the component $\heartsuit$ (rather than the component $\spadesuit$). Furthermore, the changing of $\spadesuit$ to $\heartsuit$ falsely suggests that the component $\spadesuit$ has been updated (and not deleted). This demonstrates how the artificial labelling obscures the true working of the RJMCMC algorithm.

To create a clear picture of the RJMCMC process, we will introduce the following allocation rules: first, a new component can only be born in an unoccupied position; second, the component index is set equal to the position index, for example a two-component model will have the components (w_5, μ_5, σ_5^2) and (w_7, μ_7, σ_7^2) if the 5th and the 7th position are currently taken; third, a component remains in the same position for its entire life; fourth, if a component dies, its position becomes unoccupied; fifth, if a component is born, it will be born in one of the unoccupied positions with equal probability; and finally, in the death move, the position for the death proposal is selected among the occupied positions with equal probability. Some of these rules are illustrated in the following example:

$\_\ \underline{\clubsuit}\ \_\ \_\ \_$	
$\_\ \underline{\clubsuit}\ \_\ \underline{\heartsuit}\ \_$	birth of $\heartsuit$
$\_\ \_\ \_\ \underline{\heartsuit}\ \_$	death of $\clubsuit$
$\_\ \underline{\diamondsuit}\ \_\ \underline{\heartsuit}\ \_$	birth of $\diamondsuit$

By inspection, we can easily reconstruct what happened in the RJMCMC algorithm: first, the component $\heartsuit$ was born, then the component $\clubsuit$ was deleted; and finally, the component $\diamondsuit$ was born in the position that was used by the latter $\clubsuit$. By inspection, we also see that it will be difficult to measure the quality of mixing between labels (positions) because it is not clear how to assess the gap between occupation times. We are not able to calculate the integrated autocorrelation time for the component mean μ_k since k corresponds to the position which may not always be occupied. There is however an alternative way of assessing the convergence of the algorithm with respect to the occupation of positions. The positions for birth (or death) are chosen with equal probability among the unoccupied (or occupied) positions so that each position I, $I = 1, \ldots, K_{\max}$, has in theory the same rate of occupation

given by

$$\text{occupation rate for position } I = \frac{\text{number of iterations in which } I \text{ is occupied}}{\text{total number of iterations}} \,. \tag{9.1}$$

As we do not use any other move type apart from the birth-and-death move, we cannot achieve equal occupation rates before the algorithm has converged in label switching so that we can take similar occupation rates as a sign of this convergence.

9.4.2 Proposal mechanism

We will specify the birth-and-death move with respect to the tempered posterior distribution $p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right)$ as follows. First a decision is made whether a birth or a death is attempted. If the current number I of components is greater than one and smaller than $K_{\max}$, birth and death are chosen with equal probability $q(I, I+1) = q(I, I-1) = \frac{1}{2}$. If currently either the maximum or the minimum number of components is attained, then only one move type is possible and therefore always chosen so that $q(1,2) = 1$ and $q(K_{\max}, K_{\max}-1) = 1$. Hence, the probability $q(I,J)$ of proposing a move from the I-component model to the J-component model is defined as follows:

$$q(I,J)\begin{cases} = 1 & \text{if } (I,J) \in \{(1,2), (K_{\max}, K_{\max}-1)\}\,, \\ = \frac{1}{2} & \text{if } 1 < I < K_{\max} \text{ and } J \in \{I-1, I+1\}\,, \\ = 0 & \text{otherwise.} \end{cases}$$

If a birth move is chosen, the position of the new component will be chosen from the unoccupied positions with equal probability. For simplicity and without loss of generality, let us assume that we are currently occupying the positions 1 to K in the K-component model so that we are in $x = (K, \theta) \in \Omega_K$ where

$$\theta := \left(\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, w_1, \ldots, w_{K-1}\right).$$

Note that one of the weights, here $w_K = 1 - \sum_{k=1}^{K-1} w_k$, needs to be left out in the definition of θ because it is a dummy variable. This choice may change from iteration to iteration of the RJMCMC sampler: for example, if the component containing the dummy variable is to be deleted in a death move, it will be necessary to consider another component weight dummy variable. In practice, this does not cause any problems because the implementation of the algorithm does not require an explicit declaration of the dummy variable. Similarly, the definition of the proposal state $x' = (K+1, \theta') \in \Omega_{K+1}$ omits one of the

proposed weights, here

$$w'_K = 1 - w'_{K+1} - \sum_{k=1}^{K-1} w'_k,$$

as the dummy variable by setting

$$\theta' := \left(\mu'_1, \ldots, \mu'_{K+1}, (\sigma'_1)^2, \ldots, \left(\sigma'_{K+1}\right)^2, w'_1, \ldots, w'_{K-1}, w'_{K+1}\right).$$

The new component is here assumed to be $\left(w'_{K+1}, \mu'_{K+1}, \left(\sigma'_{K+1}\right)^2\right)$ without loss of generality. To create this new component, an auxiliary random variable

$$u := \left\{w_*, \mu_*, \sigma_*^2\right\}$$

containing the weight w_*, mean μ_* and variance σ_*^2 of the new component is drawn from the corresponding prior distributions

$$\begin{aligned} w_* \Big| (K+1) &\sim \mathit{Beta}(1, K), \\ \mu_* \Big| (K+1) &\sim N(0, 1000), \\ \sigma_*^2 \Big| (K+1) &\sim \mathit{Inverse\ Gamma}(1, 1), \end{aligned}$$

which have the densities

$$\begin{aligned} p\left(w_* \middle| (K+1)\right) &= K\,(1-w_*)^{K-1}, & w_* &\in [0,1], \\ p\left(\mu_* \middle| (K+1)\right) &= (2 \cdot 1000\,\pi)^{-\frac{1}{2}} \exp\left(-\tfrac{\mu_*^2}{2 \cdot 1000}\right), & \mu_* &\in (-\infty, \infty), \\ p\left(\sigma_*^2 \middle| (K+1)\right) &= (\sigma_*)^{-2} \exp\left(-\tfrac{1}{\sigma_*^2}\right), & \sigma_*^2 &\in (0, \infty), \end{aligned}$$

and thus the joint density

$$p\left(w_*, \mu_*, \sigma_*^2 \middle| (K+1)\right) = p\left(w_* \middle| (K+1)\right) p\left(\mu_* \middle| (K+1)\right) p\left(\sigma_*^2 \middle| (K+1)\right). \tag{9.2}$$

The new component is then added to the previous configuration. While the previous component means and variances remain unchanged, their weights are multiplied by $(1 - w_*)$. The adjustment of the weights also means that the dummy variable w'_K can be derived from the previous dummy variable w_K by setting $w'_K = w_K\,(1 - w_*)$. In short, the components of the $(K+1)$-component model are given by the invertible transformation

$$\begin{aligned} w'_k &= w_k\,(1 - w_*), & k &\in \underline{K}, \\ w'_{K+1} &= w_*, & & \\ \mu'_k &= \mu_k, & k &\in \underline{K}, \\ \mu'_{K+1} &= \mu_*, & & \\ (\sigma'_k) &= \sigma_k, & k &\in \underline{K}, \\ \left(\sigma'_{K+1}\right) &= \sigma_*. & & \end{aligned}$$

For convenience, we will write (θ, u) in such a way that the grouping of variables is maintained:

$$(\theta, u) := \left(\mu_1, \ldots, \mu_K, \mu_*, \sigma_1^2, \ldots, \sigma_K^2, \sigma_*^2, w_1, \ldots, w_{K-1}, w_*\right).$$

The above invertible transformation can then be expressed by the transformation $\theta' := t(\theta, u)$ with

$$\begin{aligned} t : &\left(\mu_1, \ldots, \mu_K, \mu_*, \sigma_1^2, \ldots, \sigma_K^2, \sigma_*^2, w_1, \ldots, w_{K-1}, w_*\right) \\ &\mapsto \left(\mu_1, \ldots, \mu_K, \mu_*, \sigma_1^2, \ldots, \sigma_K^2, \sigma_*^2, w_1\left(1 - w_*\right), \ldots, w_{K-1}\left(1 - w_*\right), w_*\right). \end{aligned} \tag{9.3}$$

Note that θ' and (θ, u) are both of dimension $\left(3\left(K+1\right)-1\right)$. It remains to calculate the determinant of the Jacobian matrix of the transformation (θ, u) to θ'. As θ' and (θ, u) are of dimension $\left(3\left(K+1\right)-1\right)$, the Jacobian matrix $J = \frac{\partial\, t(\theta,u)}{\partial\,(\theta,u)}$ is of dimension $\left(3\left(K+1\right)-1\right) \times \left(3\left(K+1\right)-1\right)$. Further note that the ith component of θ' depends either solely on the ith component of (θ, u), namely when $i = 1, \ldots, \left(2\left(K+1\right)-1\right), \left(3\left(K+1\right)-1\right)$, or on the ith and the $\left(3\left(K+1\right)-1\right)$th component of (θ, u), namely when $i = \left(2\left(K+1\right)+1\right), \ldots, \left(3\left(K+1\right)-2\right)$. It follows that the lower triangle (without the diagonal) of the $\left(3\left(K+1\right)-1\right) \times \left(3\left(K+1\right)-1\right)$ Jacobian matrix $J = \frac{\partial\, t(\theta,u)}{\partial\,(\theta,u)}$ is equal to zero, i.e. $J_{ij} = \frac{\partial\,(\theta')_i}{\partial\,(\theta,u)_j} = 0$ for all i, j with $i > j$. In other words, the Jacobian matrix is an upper triangular matrix. This implies that the Jacobian determinant is equal to the product of the diagonal entries of the Jacobian matrix, i.e. $\det(J) = \prod_i J_{ii}$. As the diagonal elements are given by

$$\begin{aligned} J_{ii} &= 1 && i = 1, \ldots, \left(2\left(K+1\right)\right), \\ J_{ii} &= \left(1 - w_*\right) && i = \left(2\left(K+1\right)+1\right), \ldots, \left(3\left(K+1\right)-2\right), \\ J_{ii} &= 1 && i = \left(3\left(K+1\right)-1\right), \end{aligned}$$

the Jacobian determinant equals

$$\begin{aligned} \det(J) &= \textstyle\prod_{i=1}^{3(K+1)-1} J_{ii} \\ &= \left(1 - w_*\right)^{K-1}. \end{aligned}$$

In the reverse move from $x' = (K+1, \theta') \in \Omega_{K+1}$ with

$$\theta' := \left(\mu_1', \ldots, \mu_{K+1}', \left(\sigma_1'\right)^2, \ldots, \left(\sigma_{K+1}'\right)^2, w_1', \ldots, w_{K-1}', w_{K+1}'\right),$$

the death move is chosen with probability $q(K+1, K)$. Then one of the existing $(K+1)$ components (one of the $(K+1)$ occupied positions), here $(w_{K+1}', \mu_{K+1}', (\sigma_{K+1}')^2)$, is chosen with equal probability and its death is proposed by the inverse transformation to (9.3): first the chosen component $(w_{K+1}', \mu_{K+1}', (\sigma_{K+1}')^2)$ is removed, then the weights of the existing components

are multiplied by $\frac{1}{1-w_{K+1}}$ as they are constrained to sum to one. This yields

$$\begin{aligned}
w_* &= w'_{K+1}, \\
w_k &= \tfrac{w'_k}{1-w'_{K+1}}, && k \in \underline{K}, \\
\mu_* &= \mu'_{K+1}, \\
\mu_k &= \mu'_k, && k \in \underline{K}, \\
\sigma_* &= \sigma'_{K+1}, \\
\sigma_k &= \sigma'_k, && k \in \underline{K}.
\end{aligned}$$

From this, we obtain the component parameter vector

$$\theta = \left(\mu_1, \ldots, \mu_K, \sigma_1^2, \ldots, \sigma_K^2, w_1, \ldots, w_{K-1}\right)$$

of the proposed state $x = (K, \theta) \in \Omega_K$ and the auxiliary variable

$$u = \left\{w_*, \mu_*, \sigma_*^2\right\}.$$

We now know the proposal mechanism of the birth-and-death move and can therefore determine the acceptance probabilities (in informal notation): a birth move from $x = (K, \theta)$ to $x' = (K+1, \theta')$ is accepted with probability

$$\begin{aligned}
\alpha(x, x') &= \min\left\{1, \frac{p_\beta\left(x' \middle| \{y_j\}_{j\in\underline{n}}\right) q(K+1, K) \left|\det\left(\frac{\partial\, t(\theta,u)}{\partial\,(\theta,u)}\right)\right|}{p_\beta\left(x \middle| \{y_j\}_{j\in\underline{n}}\right) q(K, K+1)\, q(u)}\right\} \\
&= \min\left\{1, \frac{p_\beta\left(K+1, \left\{w'_k, \mu'_k, (\sigma'_k)^2\right\}_{k\in\underline{(K+1)}} \middle| \{y_j\}_{j\in\underline{n}}\right) q(K+1, K) \left|(1-w_*)^{K-1}\right|}{p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \middle| \{y_j\}_{j\in\underline{n}}\right) q(K, K+1)\, p\left(w_*, \mu_*, \sigma_*^2 \middle| (K+1)\right)}\right\} \\
&= \min\left\{1, \frac{\left[p\left(\{y_j\}_{j\in\underline{n}} \middle| K+1, \left\{w'_k, \mu'_k, (\sigma'_k)^2\right\}_{k\in\underline{(K+1)}}\right)\right]^\beta q(K+1, K)}{\left[p\left(\{y_j\}_{j\in\underline{n}} \middle| K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}}\right)\right]^\beta q(K, K+1)}\right\}.
\end{aligned} \tag{9.4}$$

To understand the cancellation step, recall that $\mu'_k = \mu_k$ and $\sigma'_k = \sigma_k$ for all

$k = 1, \ldots, K$. Hence, the ratio of posterior densities reduces to

$$\begin{aligned}
&\frac{p_\beta\left(K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}} \,\middle|\, \{y\}_{j\in\underline{n}}\right)}{p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y\}_{j\in\underline{n}}\right)} \\
&\quad = \frac{p(K+1)}{p(K)} \times \frac{p\left(\{w_k'\}_{k\in\underline{(K+1)}} \,|K+1\right)}{p\left(\{w_k\}_{k\in\underline{K}} \,|K\right)} \times \frac{\prod_{k=1}^{K+1}\left(\mu_k' \,|K+1\right)}{\prod_{k=1}^{K}\left(\mu_k \,|K\right)} \times \frac{\prod_{k=1}^{K+1}\left((\sigma_k')^2 \,|K+1\right)}{\prod_{k=1}^{K}\left(\sigma_k^2 \,|K\right)} \\
&\qquad \times \frac{\left[p\left(\{y\}_{j\in\underline{n}} \,|K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}}\right)\right]^\beta}{\left[p\left(\{y\}_{j\in\underline{n}} \,|K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}}\right)\right]^\beta} \\
&\quad = 1 \times K \times (2\cdot 1000\,\pi)^{-\frac{1}{2}} \exp\left(-\frac{\mu_*^2}{2\cdot 1000}\right) \times \left(\sigma_*^2\right)^{-1} \exp\left(-\frac{1}{\sigma_*^2}\right) \\
&\qquad \times \frac{\left[p\left(\{y\}_{j\in\underline{n}} \,|K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}}\right)\right]^\beta}{\left[p\left(\{y\}_{j\in\underline{n}} \,|K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}}\right)\right]^\beta}.
\end{aligned}$$

To simplify the ratio of the proposal densities, recall that the weight w_* lies between zero and one so that we can omit the modulus signs in $\left|(1-w_*)^{K-1}\right|$ for cancellation:

$$\begin{aligned}
&\frac{q(K+1,K)\,\left|\det\left(\frac{\partial t(\theta,u)}{(\theta,u)}\right)\right|}{q(K,K+1)\,p\left(w_*, \mu_*, \sigma_*^2 | K+1\right)} \\
&\quad = \frac{q(K+1,K)\,\left|(1-w_*)^{K-1}\right|}{q(K,K+1)\,K\,(1-w_*)^{K-1}\,(2\cdot 1000\,\pi)^{-\frac{1}{2}} \exp\left(-\frac{\mu_*^2}{2\cdot 1000}\right)\left(\sigma_*^2\right)^{-1} \exp\left(-\frac{1}{\sigma_*^2}\right)} \\
&\quad = \frac{q(K+1,K)}{q(K,K+1)\,K\,(2\cdot 1000\,\pi)^{-\frac{1}{2}} \exp\left(-\frac{\mu_*^2}{2\cdot 1000}\right)\left(\sigma_*^2\right)^{-1} \exp\left(-\frac{1}{\sigma_*^2}\right)}.
\end{aligned}$$

When multiplying the ratio of the posterior densities with the ratio of the proposal densities further cancellations occur leading to the expression given in (9.4). Similarly, the acceptance probability for the death move from x' to x is given by

$$\begin{aligned}
\alpha(x', x) &= \min\left\{1, \frac{p_\beta\left(x \,\middle|\, \{y_j\}_{j\in\underline{n}}\right) q(K,K+1)\, q(u)}{p_\beta\left(x' \,\middle|\, \{y_j\}_{j\in\underline{n}}\right) q(K+1,K)\left|\det\left(\frac{\partial\, t(\theta,u)}{\partial\,(\theta,u)}\right)\right|}\right\} \\
&= \min\left\{1, \frac{\left[p\left(\{y_j\}_{j\in\underline{n}} \,\middle|K, \left\{w_k, \mu_k, \sigma_k^2\right\}_{k\in\underline{K}}\right)\right]^\beta q(K,K+1)}{\left[p\left(\{y_j\}_{j\in\underline{n}} \,\middle|K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}}\right)\right]^\beta q(K+1,K)}\right\}.
\end{aligned}$$

Finally note that the probability of choosing a particular position for the birth proposal, which is here $\frac{1}{(K_{\max}-K)}$, or for the death proposal, which is here $\frac{1}{(K+1)}$, is not incorporated in the acceptance probability because the positioning is here a matter of computational notation and not a matter of the RJMCMC

process. However, if we want the RJMCMC process to distinguish between the different ways of allocating components, we would have to ensure that summing over all the ways of choosing K positions out of $K_{\max}$ positions gives the same posterior density as before by setting the posterior density of a particular grouping equal to $\frac{K!\,(K_{\max}-K)!}{K_{\max}!}\,p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right)$. This differentiation would also require to include the probability of selecting the position for the birth or death move in the proposal density. The ratio in the acceptance probability would thus be equal to

$$\begin{aligned}
&\frac{\frac{(K+1)!\,(K_{\max}-(K+1))!}{K_{\max}!}\,p_\beta\left(K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}} \,\middle|\, \{y\}_{j\in\underline{n}}\right)}{\frac{K!\,(K_{\max}-K)!}{K_{\max}!}\,p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right)} \\
&\quad\times \frac{q(K+1,K)\,\frac{1}{(K+1)}\,\left|\det\left(\frac{\partial t(\theta,u)}{(\theta,u)}\right)\right|}{q(K,K+1)\,p\left(w_*, \mu_*, \sigma_*^2 \middle| (K+1)\right)} \\
&\quad= \frac{p_\beta\left(K+1, \left\{w_k', \mu_k', (\sigma_k')^2\right\}_{k\in\underline{(K+1)}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right) q(K+1,K)\,\left|(1-w_*)^{K-1}\right|}{p_\beta\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k\in\underline{K}} \,\middle|\, \{y_j\}_{j\in\underline{n}}\right)\, q(K,K+1)\,p\left(w_*, \mu_*, \sigma_*^2 \middle| (K+1)\right)},
\end{aligned}$$

which is identical to the previous acceptance probability (9.4).

9.5 Running tempered transitions RJMCMC

In Chapter 7, we have seen that tempered transitions improves the mixing between modes in fixed dimension. Now we are investigating whether tempered transitions improves the mixing between dimensions, that is, in mixture modelling, the mixing between the K-component models, $K = 1, \ldots, K_{\max}$. This mixing can be measured by the integrated autocorrelation time $\tau(K)$. Furthermore, we can assess the mixing between dimensions and between labels by comparing the occupation rates (9.1) of each of the positions I, $I = 1, \ldots, K_{\max}$. When convergence is reached, the rates are approximately the same.

The tempered transitions algorithm will be based solely on birth-and-death moves (and not on any fixed-dimensional moves) at all temperature levels. To find a suitable hottest temperature, a simple RJMCMC algorithm, again solely using birth-and-death moves, was run at inverse temperatures $\beta_i = 2^{-i}$, $i = 0, 1, 2, 3$, for 100 000 iterations (not including a burn-in of 10 000 iterations). Plotting the histograms of the number of visited components reveals that the marginal distribution of K is unimodal (see Figure 9-3). We can see that, at

the hotter temperatures $\beta = \frac{1}{4}, \frac{1}{8}$, all possible K values are visited, while at the target temperature the components $K = 1, 2$ are left out. Furthermore, the heating-up causes a mode shift towards the lower number of components so that, at $\beta = \frac{1}{8}$, the most likely number of components is $K = 1$. To understand this behaviour, consider the tempered posterior distribution conditional on $K = 1$. It is of the following form

$$p_\beta\left(\mu, \sigma^2 \middle| K = 1, \{y_j\}_{j \in \underline{n}}\right) \propto p(\mu)\, p(\sigma^2) \left\{ \prod_{j=1}^{n} \exp\left[-\frac{1}{2\,\sigma^2} (y_j - \mu)^2 \right] \right\}^\beta$$
$$\propto p(\mu)\, p(\sigma^2) \exp\left[-\frac{\beta}{2\,\sigma^2} \sum_{j=1}^{n} (y_j - \mu)^2 \right].$$

This tempered posterior is identical to a non-tempered posterior modelling the data by a normal distribution with mean μ and variance $\frac{\sigma^2}{\beta}$. That means that tempered transitions fits the model $N\left(\mu, \frac{\sigma^2}{\beta}\right)$ at a particular temperature β, but only passes the values μ and σ to the next temperature level so that we have the impression that it fits the model $N(\mu, \sigma^2)$. To visualise this effect, the means $\bar{\mu}$ and $\bar{\sigma}^2$ of the μ and σ^2 samples of the conditional tempered distribution were used to define the density estimates $N\left(\bar{\mu}, \frac{\bar{\sigma}^2}{\beta}\right)$ (correct model, Figure 9-1) and $N(\bar{\mu}, \bar{\sigma}^2)$ (perceived model, Figure 9-2). The former figure (correct model) is the one that is relevant for understanding the shift to the one-component-model. As we can see, the density estimate vaguely covers the data. We can infer that the likelihood is of relatively little importance. As a result, the multimodality of the likelihood vanishes, which explains the tendency to the one-component-model. Since reducing the multimodality was the motivation for this way of tempering, this result is not surprising. On the other hand, one may expect that the marginal distribution of the components (see histogram in Figure 9-3) resembles more its uniform prior. If we expect this, we forget that the data have still some influence in the sense that there should be at least one component that has its mean within the data range to explain the observations. Another point to consider is the possibility that one data-explaining component can be replaced by two (or more) components which lie in the close neighbourhood of the replaced component and whose weights approximately sum up to the weight of the replaced component. The probability of such a replacement is however small due to the construction of the birth-and-death move. For simplicity, let us assume that there can be at most two components. Suppose that there is currently only one component and that this component lies within the data range. With probability one a birth-move will be proposed until one is successful. As the $N(0, 1000)$ prior for

the component mean is vague, it will take several attempts until a component mean within the data range is proposed. Let us assume that this proposal is accepted so that there are now two components close together. Then with probability one a death will be proposed in the next iteration. This death will have a high chance of acceptance because the surviving and the dying component have similar mean and variance values. Furthermore, the weight of the dying component is automatically absorbed by the surviving component due to the deterministic weight adjustment so that, practically, the adjusted surviving component replaces the previous configuration completely. Since a death happens almost instantly, while a birth needs time, the sampler will give more weight to the one-component model than the two-component model. This leads exactly to the behaviour we observe in the histogram of K at $\beta = \frac{1}{8}$.

From the barplots of the occupation rates at $\beta = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ (see Figure 9-4), we can infer that 100 000 iterations at $\beta = 1$ and $\beta = \frac{1}{2}$ are not enough for convergence because the occupation rates of the various positions have not reached an equal level. At $\beta = \frac{1}{4}$ and $\beta = \frac{1}{8}$, the occupation rates are quite similar. Since the hotter inverse temperature $\beta = \frac{1}{8}$ does not provide a significant reduction in the variation of occupation rates compared to $\beta = \frac{1}{4}$, the latter seems to be a sufficient hot inverse temperature so that we will set $\beta_{\min} = \frac{1}{4}$ in the tempered transitions algorithm.

As described in Chapter 7, the optimal tuning of the tempered transitions algorithm requires approximating the curve

$$g(\beta) = \mathbb{E}_{p_\beta}\left[h\left(K, \{w_k, \mu_k, \sigma_k^2\}_{k \in \underline{K}}\right)\right].$$

We can interpolate this curve by a piecewise cubic Hermite interpolation based on the anchor points $\hat{g}(\tilde{\beta}_i)$ and $\hat{g}'(\tilde{\beta}_i)$ at $\tilde{\beta}_i = 2^{-i}$, $i = 0, 1, 2$. These anchor points can be obtained by importance sampling based on the 100 000 samples at $\beta_{\min} = \frac{1}{4}$. The interpolated curve is then used in dynamic programming to search for the optimal set of inverse temperatures $\{\beta_i\}_{i=0,\ldots,n}$ minimising the sum of squares

$$S\left(\{\beta_i\}_{i=0,\ldots,n}\right) = \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left[\hat{g}(\beta_{i+1}) - \hat{g}(\beta_i)\right].$$

As the interpolated curve $\hat{g}(\beta)$ is convex, geometrically spaced temperatures are close to optimal. For comparison, $n = 30$ optimally spaced temperatures between $\beta_{\min} = \frac{1}{4}$ and $\beta_0 = 1$ yield the optimal sum of squares $S = 0.442$,

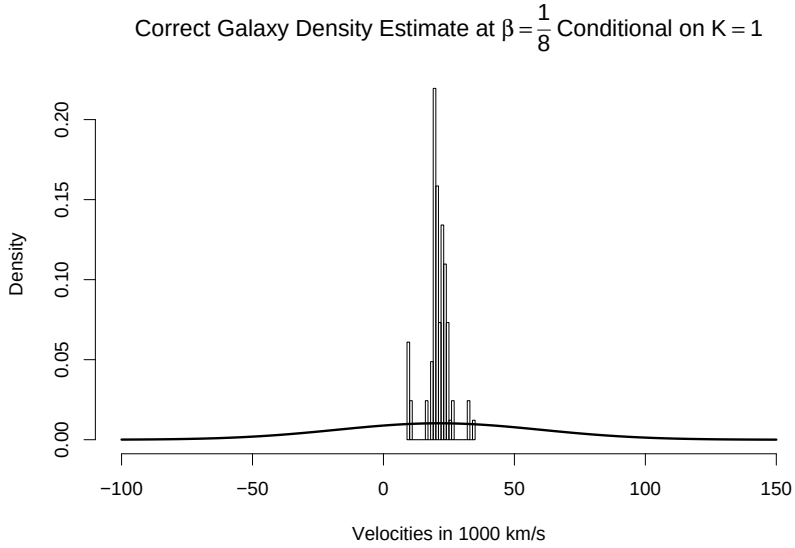

Figure 9-1: Sampling from the tempered distribution p_β conditional on $K = 1$ is equivalent to fitting a normal distribution $N\left(\mu, \frac{\sigma^2}{\beta}\right)$ to the data. This figure shows a histogram of the galaxy data overlaid by the density of the correct average model $N\left(\bar{\mu}, \frac{\bar{\sigma}^2}{\beta}\right)$.

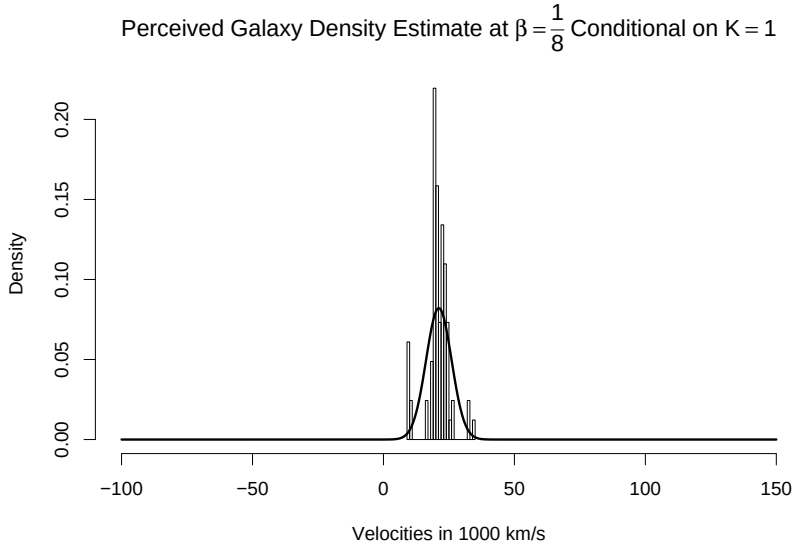

Figure 9-2: Although, at temperature β, the model $N\left(\mu, \frac{\sigma^2}{\beta}\right)$ is fitted (conditional on $K = 1$), only the μ and σ^2 values are passed on to the next temperature level so that one may have the wrong impression that actually the model $N\left(\mu, \sigma^2\right)$ is used. This figure shows a histogram of the galaxy data overlaid by the density of the perceived average model $N\left(\bar{\mu}, \bar{\sigma}^2\right)$.

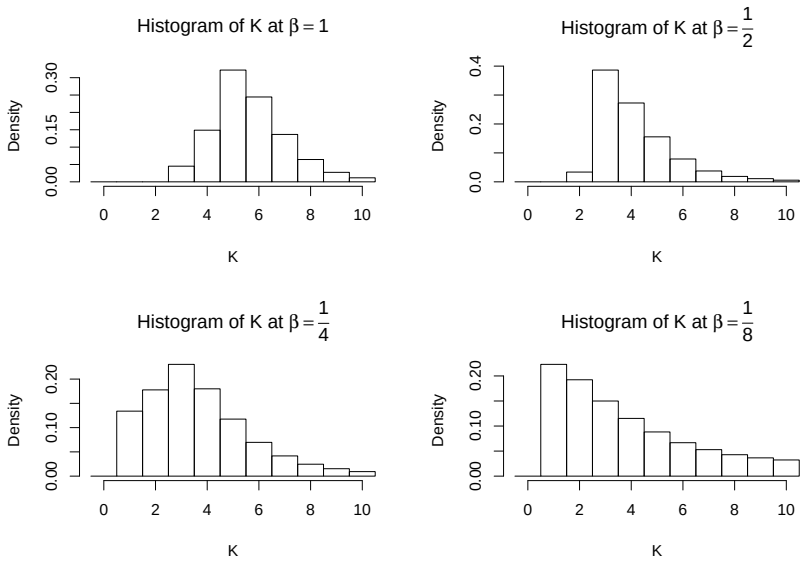

Figure 9-3: The figure shows the histograms of the variable K (component number) when sampling at the temperatures $\beta = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ by standard RJMCMC ($N = 100\,000$ iterations). As the temperature becomes hotter, the mode of the distribution shifts towards $K = 1$.

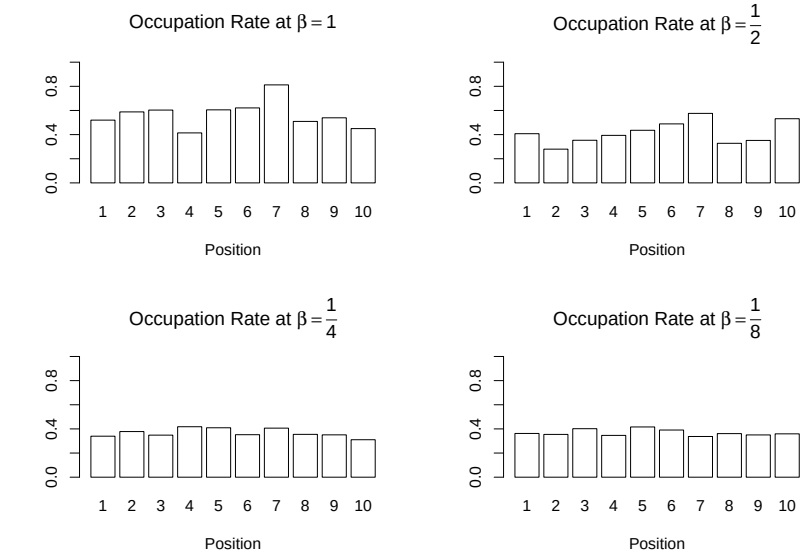

Figure 9-4: The figure shows the barplots of the occupation rates when sampling at the temperatures $\beta = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}$ by standard RJMCMC ($N = 100\,000$ iterations). Since similar occupation rates signal convergence in label switching, that this convergence has not been reached in the cases $\beta = 1$ and $\beta = \frac{1}{2}$.

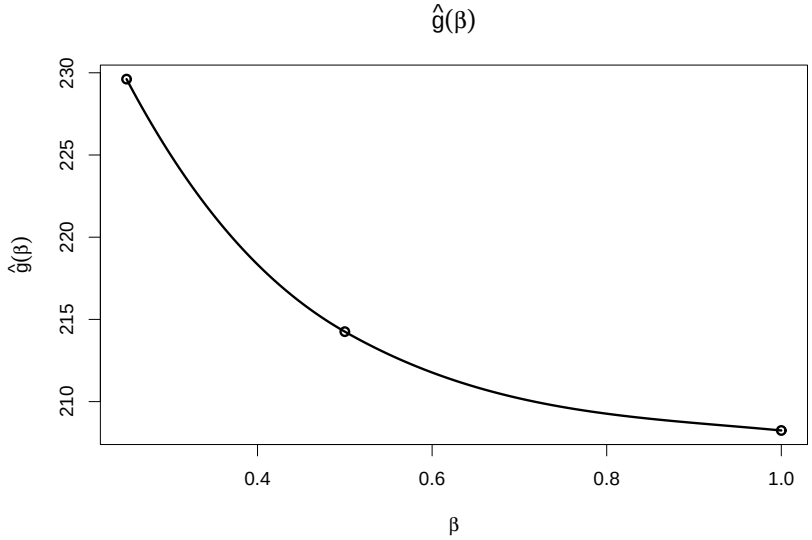

Figure 9-5: The figure shows the interpolated curve $g(\beta)$ for the variable-dimensional mixture model.

while geometrically spaced temperatures lead to the insignificantly greater $S = 0.448$. Note that no other values for n were tried because the tempered transitions run based on $n = 30$ optimal temperatures gave satisfactory results. The results will be presented in comparison with standard RJMCMC.

First note that standard RJMCMC is much slower in convergence. We can see in Figure 9-6 that it takes a long time until the occupation rates attain an approximately equal level. It therefore had to be run for $2 \cdot 10^6$ iterations (after a burn-in of 10 000) to reach convergence, while tempered transitions only required $2 \cdot 10^5$ iterations (after a burn-in of 10 000) for convergence. Despite the smaller sample size, the variability between occupation rates is noticeable smaller in the tempered transitions run (see Figure 9-7). Another sign of convergence is that the histograms of K obtained by both methods are quite similar to each other (again see Figure 9-7). There is a big difference in the computational cost of both methods: the standard RJMCMC run took one hour, while tempered transitions needed seven hours. The mixing between dimensions was assessed by the estimated integrated autocorrelation time $\hat{\tau}$ (2.3) with respect to the number K of components. The large sample size of the standard RJMCMC run caused storage problems when computing $\hat{\tau}(K)$ because the estimator is a window estimator requiring the entire sample to be in the active memory which could only store $5 \cdot 10^5$ iterations. Since the accuracy

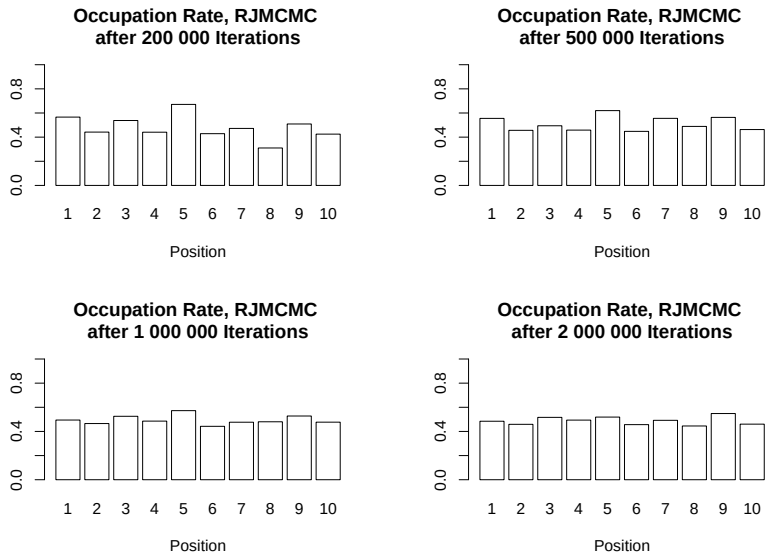

Figure 9-6: The figure shows the occupation rates after $N = 2 \cdot 10^5, 5 \cdot 10^5, 1 \cdot 10^6, 2 \cdot 10^6$ iterations when sampling from the target distribution by standard RJMCMC. The variability in the occupation rates in the first three cases indicates that convergence in label switching has not been reached. For convergence, we need approximately $2 \cdot 10^6$ iterations.

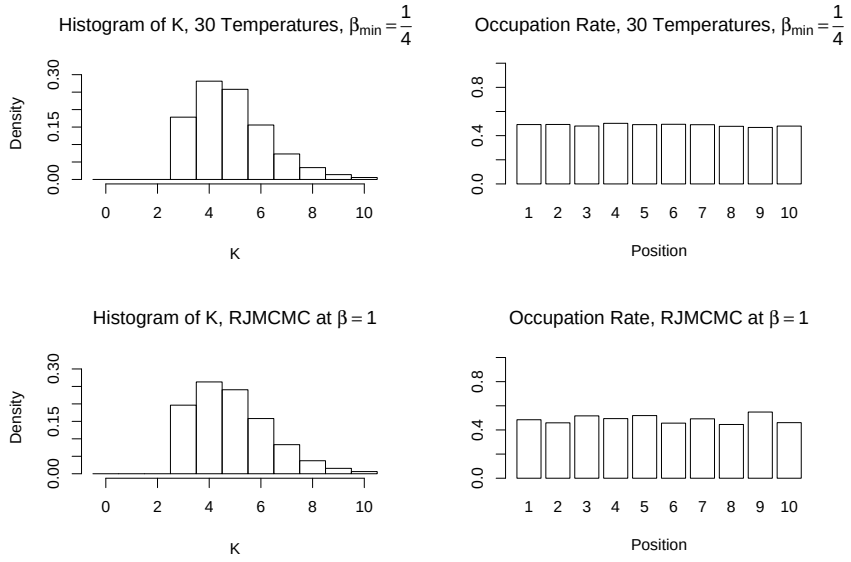

Figure 9-7: For comparison, the histograms for K and the occupation rates are plotted for the standard RJMCMC run ($N = 2\,000\,000$ iterations) and the tempered transitions run ($N = 2\,000\,000$ iterations, $n = 30$ optimal temperatures between $\beta_{\min} = \frac{1}{4}$ and $\beta_0 = 1$). Both methods converge. Standard RJMCMC shows more variability in the occupation rates despite its larger sample size.

of the estimator increases with the accuracy of the sample mean of K, first the sample mean was computed based on the entire sample size, then the estimated autocorrelation time was calculated based on the first $5 \cdot 10^5$ iterations (after burn-in) of the run, but with respect to the sample mean of the $2 \cdot 10^6$ iterations. Due to the smaller sample size, the integrated autocorrelation time of the tempered transitions RJMCMC run could be determined based on the entire sample size of $2 \cdot 10^5$. The results are that tempered transitions ($\hat{\tau}(K) = 355.5$) is 15 times more accurate than standard RJMCMC ($\hat{\tau}(K) = 5435.8$). In this example, the higher computational cost of tempered transitions is outweighed by its superior mixing between dimensions.

To summarise, we have verified that tempered transitions can also be applied to trans-dimensional problems. It is possible to mix fixed-dimensional and trans-dimensional move types. However, in our application, only trans-dimensional move types were incorporated to investigate their impact on mixing in isolation. In our example, trans-dimensional tempered transitions performed better than standard RJMCMC.

Chapter 10

Conclusions

Let us close this research on "Mode Jumping in MCMC" with summarising and discussing its key issues and results. First, we introduced the theory behind MCMC and explained the mode jumping problem. The latter arises because either MCMC cannot reach isolated modes at all, for example when the states are updated component-wise and the modes do not lie in line with the direction of the update, or it cannot find any other isolated mode in a reasonable amount of time, for example when components are updated jointly and the probability of hitting another mode is tiny under the proposal distribution. We then discussed the main ideas behind the existing mode jumping approaches. We found that there are quite a few tempering methods that have been reported as tackling the mode jumping problem well. These methods use a sequence of auxiliary distributions between the target distribution, at which standard MCMC mixes poorly, and the hottest distribution, at which standard MCMC mixes fast. The fast mixing is possible because the hottest distribution features less definite modes than the target distribution. These modes also occupy a larger space so that they can be more easily hit by a standard proposal mechanism. Among the tempering methods, we picked Neal's (1996) tempered transitions because it has the advantage that it is a single-chain method which does not require estimating the normalisation constants of the auxiliary distributions. As the idea of tempering comes from stochastic optimisation, we were also interested in comparing tempered transitions to another method, mode jumping via local optimisation (Tjelmeland and Hegstad 2001), that is based on deterministic mode searches. We chose the toy example that Tjelmeland and Hegstad (2001) used to show the benefits of their method. In that toy example, we observed that their method has difficulties escaping modes closely surrounded by other modes which slows down the mixing of

the sampler. Tempered transitions on the other hand needs on average the same number of attempts to leave a mode independent of its neighbourhood structure. Moreover, tempered transitions was easier to implement and twice as fast as the other method in the toy example. Seeing the power of tempered transitions and its advantages on the one hand and its computational cost on the other hand, it was investigated how tempered transitions could be made more efficient. The efficiency depends on the inverse temperatures $\{\beta_i\}_{i=0}^n$ defining the auxiliary distributions $p_{\beta_i}(x) \propto \pi(x) \exp\left[-\beta_i\, h(x)\right]$, $i = 0, 1, \ldots, n$, and the Markov transition kernels $\{T_{\beta_i}\}_{i=1}^n$ used to create the proposal state in tempered transitions. In particular, the hottest temperature $\beta_{\min}$ and the corresponding transition kernel $T_{\beta_{\min}}$ are essential for the efficiency of tempered transition. If $\beta_{\min}$ is not hot enough or $T_{\beta_{\min}}$ too slow in mixing, then tempered transitions will mix poorly. On the other hand, if $\beta_{\min}$ is far hotter than necessary, then more intermediate distributions between the hottest and the target distribution are required to obtain reasonable acceptance rates so that the algorithm is not cost-efficient. Given the hottest temperature $\beta_{\min}$, a fast mixing kernel $T_{\beta_{\min}}$ and the number n of temperatures, the sequence $\{\beta_i\}_{i=0}^n$ and the kernels $\{T_{\beta_i}\}_{i=1}^n$ are ideally chosen such that the expected acceptance probability $\mathbb{E}_\varphi(\alpha)$ where φ denotes the distribution of the secondary chain $(X_0, \ldots, X_n, X'_{n-1}, \ldots, X'_0)$ generated by the proposal mechanism is maximised. We can then compare the minimal expected acceptance probabilities for several values of n and use the most cost-efficient n as the final value. As this true optimisation problem is usually intractable, we suggested tackling it implicitly by finding a solution to the related problem of minimising the sum of squares

$$\mathbb{E}_\varphi \left\{ \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left[h(X'_i) - h(X_i) \right] \right\}$$

again ideally by the choice of $\{\beta_i\}_{i=0}^n$ and $\{T_{\beta_i}\}_{i=1}^n$, but more realistically by the idealising assumption that the transition kernels generate independent samples from the tempered distribution with respect to which they satisfy detailed balance. Under the idealising assumption, the problem simplifies to optimising the sum of squares

$$\begin{aligned} S\left(\{\beta_i\}_{i=0}^n\right) &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left\{ \mathbb{E}_\varphi \left[h(X'_i) \right] - \mathbb{E}_\varphi \left[h(X_i) \right] \right\} \\ &= \sum_{i=0}^{n-1} (\beta_i - \beta_{i+1}) \left[g(\beta_{i+1}) - g(\beta_i) \right] \end{aligned}$$

where $g(\beta) = \mathbb{E}_{p_\beta}[h(X)]$ by the choice of temperatures. We derived that $g(\beta)$ is decreasing with derivative $g'(\beta) = -\mathrm{var}_{p_\beta}[h(X)]$. Using this property, we proved that the solution to the problem satisfies the ordering constraint $\beta_{\min} = \beta_n < \cdots < \beta_1 = \beta_0$ where the doubling of temperatures $\beta_1 = \beta_0$ was introduced to ease the presentation of tempered transitions in general. (If the doubling constraint is relaxed, then the best scheme will be $\beta_{\min} = \beta_n < \cdots < \beta_1 < \beta_0$.) We also showed that geometric temperatures are optimal if the curve has the same shape as the model curve $g(\beta) = \frac{1}{2\beta}$. It was also suggested to use either simulated annealing or dynamic programming to solve the problem numerically. In the simplified Witch's Hat toy example, we demonstrated that the curve $g(\beta)$ can be very different from the model curve, in which case the geometric schedule is far from optimal. We also discussed that an appropriate temperature sequence will place more temperatures in areas where $g(\beta)$ is strongly decaying than in others. The simplified Witch's Hat is an excellent toy problem because the true optimisation problem of maximising $\mathbb{E}_\varphi(\alpha)$ can actually be tackled under the "ideal world" assumption that the auxiliary states of the secondary chain are independent samples from the tempered distributions with respect to which they were produced. We could thus see that, in this example, the optimal solutions of the true and of the related problem lie close together independent of the shape of $g(\beta)$. Furthermore, running the "ideal world" temperatures in various "real world" scenarios in a case where geometric spacing was not optimal showed that the "ideal world" optimisation also improves the performance under "real world" conditions. Another result was that the temperature sequence had a greater impact on the mixing than the step size plan defining the Markov transition kernels. All these findings encouraged the further development of the tuning technique for real world applications and in particular for fixed-dimensional mixture modelling. We have for example to be careful when defining the tempered distributions because not all tempered distributions are proper. We could prove however that it is safe to temper any posterior distribution by tempering the likelihood contribution, but not the prior. The core problem in complex applications is that the curve $g(\beta)$ is unknown and needs to be approximated for the optimisation. We suggested to use the same sample from the hottest distribution to estimate $g(\beta) = \mathbb{E}_{p_\beta}[h(x)]$ and $g'(\beta) = -\mathrm{var}_{p_\beta}[h(X)]$ for some β values by importance sampling and then to interpolate the curve between these anchor points. This interpolation proved to be robust. The slight variability did not have a significant effect on the final set of inverse temperatures. The

importance sampling step requires very little effort. The sample from the hottest distribution usually exists anyway because we have to test the mixing at the hottest temperature. The importance sampling code can be written in such a way that it can be used for several applications of different dimension because it suffices to know the output chain $\left\{h\left(X^{(t)}\right)\right\}_{t=1}^{N}$ which always takes univariate values. From this chain, the importance weights can be deduced thanks to the special tempering structure. Similarly, the code for the interpolation and the optimisation can be written problem-independent because the anchor points and the resulting interpolation are also always univariate. However, if we want to save some programming effort, we can also choose the temperatures by a rule of thumb. If we simply plot the anchor points (without interpolation), we can already recognise the shape of $g(\beta)$ and pick a sensible, albeit not optimal temperature rule: if the curve is almost a straight line, a linear spacing is appropriate; if it is clearly convex, then we can use a geometric spacing; and if it is clearly concave, we need an anti-geometric spacing. In our example, the optimisation showed that the geometric scheme was close to optimal. In consequence, both the optimal and the geometric schedule performed similarly well in the final tempered transitions run. The fixed-dimensional mixture modelling problem is a good example for the power of tempered transitions. The mode jumping method succeeded in mixing between permutations of the model (label-switching) while standard MCMC failed completely.

Since we often use variable-dimensional mixture models to learn about the uncertainty surrounding the number of mixture components, we were also keen on investigating whether tempered transitions can be applied to variable-dimensional problems and, if so, whether it helps the mixing between dimensions. To lay the foundations, we explained the theory behind standard RJMCMC in more detail than the original paper did, and presented some further developments of RJMCMC, among them some tempering methods that were all reported to work well. We then verified that tempered transitions can be combined with RJMCMC without further adjustments, and applied the combined method to variable-dimensional mixture modelling. As we only implemented trans-dimensional move types (and no fixed-dimensional ones), the improvement in mixing that we observed in the example in comparison to standard RJMCMC was obviously due to a better mixing between dimensions (and not within a particular dimension).

In conclusion, tempered transitions is a powerful mode jumping method in fixed and variable dimension. The developed tuning technique works well and leads to a better temperature choice than geometric temperatures when the shape of $g(\beta)$ differs significantly from that of the geometric model curve $g(\beta) = \frac{1}{2\beta}$.

We have seen that it is possible to design a tuning technique for tempered transitions. It would be interesting to see whether similar ideas can be applied to optimise other tempering methods, such as simulated tempering, Metropolis-coupled MCMC, or the equi-energy sampler. If such ideas are developed, they may be tested on the simplified Witch's Hat because the direct computation of expected acceptance probabilities is there feasible. Similar investigations may also help in designing a temperature schedule for simulated annealing that allows fast convergence to the global optima.

We have also discussed that, so far, there are no reliable convergence diagnostics because they cannot tell whether all the modes of the distribution have been found by the MCMC runs. It would be interesting to find out how far the information gained at hotter temperatures can help in convergence diagnosis. By plotting the histograms of the random variables, we may for example learn about the approximate region of the target modes. Perhaps, there exists some variable whose sample mean only converges to the right value if the MCMC method mixes well between all the modes of the target distribution. If such a variable exists, we could try to estimate it by importance sampling. If we are successful, we can compare this importance estimate with the MCMC estimate obtained by sampling from the target distribution. If both estimates are similar, we can assume that the MCMC method has converged at the target temperature.

References

Al-Awadhi, F., Hurn, M. and Jennison, C. (2004a). Improving the acceptance rate of reversible jump MCMC proposals, *Statistics and Probability Letters* **69**(2): 189–198.

Al-Awadhi, F., Hurn, M. and Jennison, C. (2004b). Statistical image analysis for a confocal microscopy two-dimensional section of cartilage growth, *Applied Statistics* **53**: 31–49.

Atchadé, Y. F. and Rosenthal, J. S. (2005). On adaptive Markov chain Monte Carlo algorithms, *Bernoulli* **11**(5): 815–828.

Bauer, H. (2001). *Measure and Integration Theory*, Walter de Gruyter, New York.

Bertrand, C., Ohmi, M., Suzuki, R. and Kado, H. (2001). A probabilistic solution to the MEG inverse problem via MCMC methods: The reversible jump and parallel tempering algorithms, *IEEE Transactions on Biomedical Engineering* **48**(5): 533–542.

Besag, J. and Green, P. J. (1993). Spatial statistics and Bayesian computation (with discussion), *Journal of the Royal Statistical Society B* **55**(1): 25–37, 53–102.

Besag, J., Green, P., Higdon, D. and Mengersen, K. (1995). Bayesian computation and stochastic systems, *Statistical Science* **10**(1): 3–66.

Braak, ter, C. J. F. (2006). A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces, *Statistics and Computing* **16**(3): 239–249.

Brooks, S. P. and Roberts, G. O. (1998). Convergence assessment techniques for Markov chain Monte Carlo, *Statistics and Computing* **8**: 319–335.

Brooks, S. P., Fan, Y. and Rosenthal, J. S. (2006). Perfect forward simulation via simulated tempering, *Communications in Statistics - Simulation and Computation* **35**(3): 683–713.

Brooks, S. P., Giudici, P. and Roberts, G. O. (2003). Efficient construction of reversible jump Markov chain Monte Carlo proposal distribution (with discussion), *Journal of the Royal Statistical Society B* **65**: 3–55.

Burden, R. L. and Faires, J. D. (2001). *Numerical Analysis*, 7th edn, Brooks-Cole, Pacific Grove, California.

Cappé, O., Robert, C. P. and Rydén, T. (2003). Reversible jump, birth-and-death and more general continuous time Markov chain Monte Carlo samplers, *Journal of the Royal Statistical Society B* **65**(3): 679–700.

Celeux, G., Hurn, M. and Robert, C. P. (2000). Computational and inferential difficulties with mixture posterior distributions, *Journal of the American Statistical Association* **95**(451): 957–970.

Chan, K. S. and Geyer, C. J. (1994). Discussion on Markov chains for exploring posterior distributions (by L. Tierney), *The Annals of Statistics* **22**(4): 1747–1758.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference, *The Annals of Statistics* **32**(6): 2385–2411.

Corcoran, J. N. and Tweedie, R. L. (2002). Perfect sampling from independent Metropolis-Hastings chains, *Journal of Statistical Planning and Inference* **104**: 297–314.

Cowles, M. K. and Carlin, B. P. (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review, *Journal of the American Statistical Association* **91**(434): 883–904.

Damien, P., Wakefield, J. and Walker, S. (1999). Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables, *Journal of the Royal Statistical Society B* **61**(2): 331–344.

Del Moral, P., Doucet, A. and Jasra, A. (2006). Sequential Monte Carlo samplers, *Journal of the Royal Statistical Society B* **68**(3): 411–436.

Dellaportas, P. and Papageorgiou, I. (2006). Multivariate mixtures of Normals with unknown number of components, *Statistics and Computing* **16**: 57–68.

Doucet, A., Godsill, S. and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing* **10**(3): 197–208.

Eberle, A. and Marinelli, C. (2006). Stability of sequential Markov chain Monte Carlo methods, *Technical report*, Institut für Angewandte Mathematik, Universität Bonn. Available under http://www-wt.iam.uni-bonn.de/∼eberle/OxfordProceedings2810.pdf.

Edwards, R. G. and Sokal, A. D. (1988). Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm, *Physical Review D* **38**(6): 2009–2012.

Franconi, L. and Jennison, C. (1997). Comparison of a genetic algorithm and simulated annealing in an application to statistical image reconstruction, *Statistics and Computing* **7**: 193–207.

Friel, N. and Pettitt, A. N. (2008). Marginal likelihood estimation via power posteriors, *Journal of the Royal Statistical Society B* **70**(3): 589–607.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* **85**(410): 398–409.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion), *Statistical Science* **7**(4): 457–511.

Gelman, A., Carlin, J. B., Stern, H. S. and Rubin, D. B. (2004). *Bayesian Data Analysis*, second edn, Chapman & Hall, London.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**(6): 721–741.

Geyer, C. J. (1991). Markov chain Monte Carlo maximum likelihood, *Computer Science and Statistics* **23**: 156–163.

Geyer, C. J. (1992). Practical Markov chain Monte Carlo (with discussion), *Statistical Science* **7**(4): 473–511.

Geyer, C. J. and Thompson, E. A. (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference, *Journal of the American Statistical Association* **90**(431): 909–920.

Gilks, W. R., Roberts, G. O. and George, E. I. (1994). Adaptive direction sampling, *The Statistician* **43**(1): 179–189.

Gilks, W. R., Roberts, G. O. and Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo through regeneration, *Journal of the American Statistical Association* **93**(443): 1045–1054.

Goswami, G. and Liu, J. S. (2007). On learning strategies for evolutionary Monte Carlo, *Statistics and Computing* **17**(1): 23–38.

Gramacy, R. B., Samworth, R. J. and King, R. (2007). Importance tempering, *Technical report*, Statistical Laboratory, University of Cambridge. Available under www.arxiv.org/abs/0707.4242v4.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination, *Biometrika* **82**(4): 711–732.

Green, P. J. (2003). Trans-dimensional Markov chain Monte Carlo, *in* P. J. Green, N. L. Hjort and S. Richardson (eds), *Highly Structured Stochastic Systems*, Oxford University Press, chapter 6, pp. 179–198.

Green, P. J. and Han, X.-L. (1992). Metropolis methods, Gaussian proposals, and antithetic variables, *Lecture Notes in Statistics* **74**: 142–164.

Grimmett, G. R. and Stirzaker, D. R. (2004). *Probability and Random Processes*, third edn, Oxford University Press, Oxford.

Hajek, B. (1988). Cooling schedules for optimal annealing, *Mathematics of Operations Research* **13**(2): 311–329.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**(1): 97–109.

Higdon, D. (1998). Auxiliary variable methods for Markov chain Monte Carlo with applications, *Journal of the American Statistical Society* **93**(442): 585–595.

Hurn, M. A., Rue, H. and Sheehan, N. A. (1999). Block updating in constrained Markov chain Monte Carlo sampling, *Statistics & Probability Letters* **41**: 353–361.

Jasra, A., Holmes, C. C. and Stephens, D. A. (2005). Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling, *Statistical Science* **20**(1): 50–67.

Jasra, A., Stephens, D. A. and Holmes, C. C. (2007a). On population-based simulation for static inference, *Statistics and Computing* **17**(3): 263–279.

Jasra, A., Stephens, D. A. and Holmes, C. C. (2007b). Population-based reversible jump Markov chain Monte Carlo, *Biometrika* **94**(4): 787–807.

Jennison, C. (1993). Discussion on the meeting on the Gibbs sampler and other Markov chain Monte Carlo methods, *Journal of the Royal Statistical Society B* **55**(1): 53–56.

Jennison, C. and Sheehan, N. (1995). Theoretical and empirical properties of the genetic algorithm as a numerical optimizer, *Journal of Computational and Graphical Statistics* **4**(4): 296–318.

Johnson, V. E. (1996). Studying convergence of Markov chain Monte Carlo algorithms using coupled sample paths, *Journal of the American Statistical Association* **91**(433): 154–166.

Johnson, V. E. (1998). A coupling-regeneration scheme for diagnosing convergence in Markov chain Monte Carlo algorithms, *Journal of the American Statistical Association* **93**(441): 238–248.

Kirkpatrick, S., Gelatt, Jr., C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing, *Science* **220**(4598): 671–680.

Kou, S. C., Zhou, Q. and Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics (with discussion), *The Annals of Statistics* **34**(4): 1581–1652.

Liang, F. and Wong, W. H. (2001). Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models, *Journal of the American Statistical Association* **96**(454): 653–666.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* **93**(443): 1032–1044.

Liu, J. S., Liang, F. and Wong, W. H. (2000). The multiple-try method and local optimization in Metropolis-sampling, *Journal of the American Statistical Association* **95**(449): 121–134.

Marinari, E. and Parisi, G. (1992). Simulated tempering: a new Monte Carlo scheme, *Europhysics Letters* **19**(6): 451–458.

Matthews, P. (1993). A slowly mixing Markov chain with implications for Gibbs sampling, *Statistics and Probability Letters* **17**: 231–236.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines, *The Journal of Chemical Physics* **21**(6): 1087–1092.

Mira, A. and Tierney, L. (2002). Efficiency and convergence properties of slice samplers, *Scandinavian Journal of Statistics* **29**: 1–12.

Mira, A., Møller, J. and Roberts, G. O. (2001). Perfect slice samplers, *Journal of the Royal Statistical Society B* **63**(3): 593–606.

Murdoch, D. J. and Green, P. J. (1998). Exact sampling from a continuous state space, *Scandinavian Journal of Statistics* **25**(3): 483–502.

Neal, R. M. (1996). Sampling from multimodal distributions using tempered transitions, *Statistics and Computing* **6**: 353–366.

Neal, R. M. (2001). Annealed importance sampling, *Statistics and Computing* **11**: 125–139.

Neal, R. M. (2003). Slice sampling (with discussion), *The Annals of Statistics* **31**(3): 705–767.

Nummelin, E. (1984). *General irreducible Markov chains and non-negative operators*, Cambridge University Press, Cambridge.

Peskun, P. H. (1973). Optimum Monte-Carlo sampling using Markov chains, *Biometrika* **60**(3): 607–612.

Priestley, M. B. (1981). *Spectral Analysis and Time Series*, Academic Press, London.

Propp, J. G. and Wilson, D. B. (1996). Exact sampling with coupled Markov chains and applications to statistical mechanics, *Random Structures Algorithms* **9**(1-2): 223–252.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components (with discussion), *Journal of the Royal Statistical Society B* **59**(4): 731–792.

Ripley, B. D. (1987). *Stochastic Simulation*, John Wiley & Sons, New York.

Robert, C. P. (1994). Discussion on Markov chains for exploring posterior distributions (by L. Tierney), *The Annals of Statistics* **22**(4): 1742–1747.

Robert, C. P. (1996). Mixtures of distributions: inference and estimation, *in* W. R. Gilks, S. Richardson and D. J. Spiegelhalter (eds), *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, pp. 441–464.

Robert, C. P. and Casella, G. (1999). *Monte Carlo statistical methods*, Springer, New York.

Roberts, G. O. and Rosenthal, J. S. (1999). Convergence of slice sampler Markov chains, *Journal of the Royal Statistical Society B* **61**(3): 643–660.

Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithms, *Statistical Science* **16**(4): 351–367.

Roberts, G. O. and Rosenthal, J. S. (2002). The polar slice sampler, *Stochastic Models* **18**(2): 257–280.

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies, *Journal of the American Statistical Association* **85**(411): 617–624.

Sharp, R. M. (2003). *An Approximate Alternative To Perfect Simulation*, PhD thesis, University of Bath.

Stander, J. and Silverman, B. W. (1994). Temperature schedules for simulated annealing, *Statistics and Computing* **4**: 21–32.

Stephens, M. (2000). Bayesian analysis of mixture models with an unknown number of components – an alternative to reversible jump methods, *The Annals of Statistics* **28**(1): 40–74.

Swendsen, R. H. and Wang, J.-S. (1987). Nonuniversal critical dynamics in Monte Carlo simulations, *Physical Review Letters* **58**(2): 86–88.

Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion), *The Annals of Statistics* **22**(4): 1701–1762.

Tierney, L. (1996). Introduction to general state-space Markov chain theory, *in* W. R. Gilks, S. Richardson and D. J. Spiegelhalter (eds), *Markov Chain Monte Carlo in Practice*, Chapman & Hall, London, pp. 59–74.

Tierney, L. (1998). A note on Metropolis-Hastings kernels for general state spaces, *The Annals of Applied Probability* **8**(1): 1–9.

Tjelmeland, H. (2005). Personal communication.

Tjelmeland, H. and Eidsvik, J. (2004). On the use of local optimizations within Metropolis-Hastings updates, *Journal of the Royal Statistical Society B* **66**: 411–427.

Tjelmeland, H. and Hegstad, B. K. (2001). Mode jumping proposals in MCMC, *Scandinavian Journal of Statistics* **28**: 205–223.

Walsh, G. R. (1975). *Methods of Optimization*, John Wiley & Sons, London.